



Teknisk rapport om dato- og tidsberegninger

Jens Peter Søtoft

Technical Report no. 05/02
ISSN: 0107-8283

HCØ Tryk

University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

Kildetekst: D:\kalrap\kalrap200503.doc
De tilhørende tabeller ligger i underbiblioteket \tabeller\

Jens Peter Søltoft, november 2004, revideret marts 2005

Indholdsfortegnelse

Indholdsfortegnelse	3
Sammendrag	5
Forord	7
1. Et rids af den vestlige kalenders historie	9
Kalendertyper	9
Cæsars reform - den julianske kalender	10
Status i dag	14
2. Kontrol af en dato	15
Kort introduktion til algoritmerne	15
Hvornår skal en dato kontrolleres	15
2 algoritmer: Skudårsundersøgelse og datokontrol	16
Algoritme 1: ÅRTYPE, Danmark og Norge	17
Algoritme 2: VALIDATO, Danmark og Norge	18
3. Grundformler til kalenderberegninger	19
Heltalsregning	19
4. Interkalation - definition og eksempler	23
Egenskaber ved kalenderberegninger	23
Enhed, periode, cyklus	23
Interkalation	23
Formler for jævne interkalationer	24
Eksempler på interkalation	27
Algoritme 3: DAGIÅR, Danmark og Norge	31
Algoritme 4: DATOIÅR, Danmark og Norge	31
5. Den julianske dagtælling	33
At regne i antal dage	33
Den julianske dagtælling og dens historie	33
Algoritme 5: JDAG, Danmark og Norge	37
Algoritme 6: DATOJ, Danmark og Norge	38
Antal dage	39
Algoritme 7: DATODIFF, Danmark og Norge	39
Algoritme 8: NYDATO, Danmark og Norge	39
Ny omsætning mellem dato og dagnummer i et år	40
Algoritme 3.2, DAGIÅR, Danmark og Norge	40
Algoritme 4.2, DATOIÅR, Danmark og Norge	40
6. Ugenummer og ugedag	41
Definition af ugenumrene	41
Ugebasisdato for et år	43
Algoritme 9, UGE, Danmark og Norge	45
Algoritme 10, UGESTART, Danmark og Norge	46
7. Beregningen af Påskedag	47
Placeringen af Påskedag	47
Juliansk Påskedag: 19-årig månecyklus og 28-årig ugedagscyklus	48
Gregoriansk Påskedag: Korrektioner med sol- og månejævning	50
Algoritme 11, PÅSKEDAG, Danmark og Norge	54

Indholdsfortegnelse

8. Klassifikation af en dato	55
Nationale helligdage	55
Algoritme 12, DAGTYPE, Danmark	57
Undersøgelse af et datointerval i Danmark	59
Algoritme 13, ARBDIFF, Danmark	59
9. Eksempler på datoberegninger	65
Eksempel 1: Find datoen for evt. søndag mellem jul og nytår	65
Eksempel 2: Find 1. torsdag i en måned	65
Eksempel 3: Find 1. fredag efter 3. tirsdag i en måned	66
Eksempel 4: Find startdato for en uge, der indeholder en given dato	66
Eksempel 5: Beregning af forfaldsdato (bankdag)	67
10. Beregning af tidspunkter	69
Underinddelinger af døgn	69
Algoritme 14, UPAKDDGN, generel	71
Algoritme 15, JTID, generel	71
Algoritme 16, TIDJ, generel	72
Algoritme 17, TIDDIFF, generel	72
Algoritme 18, NYTID, generel	73
11. Eksempler på algoritmer for andre lande	75
Nationale forhold - historie og politik har stor indflydelse	75
Algoritme 1S, ÅRTYPE, Sverige	75
Algoritme 2S, VALIDATO, Sverige	76
Algoritme 1R, ÅRTYPE, Romersk-katolske kirke	78
Algoritme 2R: VALIDATO, Romersk-katolske kirke	79
Algoritme 1U, ÅRTYPE, Storbritannien	80
Algoritme 2U: VALIDATO, Storbritannien	81
Algoritme 5G: JDAG, generel	82
Algoritme 6G: DATOJ, generel	83
Tillæg A. Litteraturhenvisninger	85
Tillæg B. Liste over kronologiske begreber	87
Tillæg C. Oversættelse af bullen: ”Inter Gravissimas”	91
Indledende Bemærkninger	91
Fortale til bullen	91
Inter Gravissimas	91
Supplerende deklaration for dem, der ikke gik over i 1582	95
Tillæg D. Forordning om gregoriansk kalender i Danmark	97
Tillæg E. Overgang fra juliansk til gregoriansk kalender	99

Sammendrag

Denne rapport er resultatet af en årelang undersøgelse af dato- og tidsberegninger og indeholder en historisk baggrund for vores kalender, matematiske bevisførelser for algoritmer og endelig algoritmerne selv.

I første række er rapporten rettet mod programmører, som ønsker nogle brugbare og effektive algoritmer til denne type beregninger.

Algoritmerne er præsenteret som generel Algol/Pascal-kode, og kan omskrives til et almindeligt 3.generationssprog på det anlæg, der aktuelt er til rådighed.

Kontrol af datoers validitet bliver diskuteret, og det påpeges, at en sådan kontrol kun er nødvendig i nogle situationer.

Sammen med de matematiske bevisførelser og præsentation af algoritmerne er der lagt stor vægt på at forklare og afdække de historiske hændelsesforløb og årsager til at vores kalender har fået det udseende, som den har nu.

Astronomiske og kronologiske begreber er markeret med * og forklaret i et minileksikon i Tillæg B.

I teksten er der indsat en lang række oversigtstabeller, så man også ad den vej kan se sammenhænge mellem formler og data.

Den matematiske kerne i rapporten består af to ting:

- en diskussion af de forskellige elementære operatorer og deres indbyrdes sammenhæng. Her tænkes specielt på heltalsdivision, modulus, heltalsdel af et tal og operatorernes opførsel for positive og negative værdier. Det påvises, at så længe man holder sig til ikke-negative værdier, vil de 2 forskellige definitioner for heltalsdivision og modulus være ens.
- indførelse af begreberne regelmæssig* og jævn interkalation* sammen med en præsentation af og bevisførelse for 3 sammenhørende formler, der kan håndtere jævne interkalationer, når der eksisterer et ikke-heltalligt forhold mellem de forskellige tidsenheder. Derpå demonstreres de 3 formlers generelle anvendelighed i mange forskellige problemstillinger indenfor dato- og tidsberegninger.

Rapporten koncentrerer sig om den danske kalender, mens algoritmer til andre landes kalendere er perifert omtalt i kapitel 11.

De 5 tillæg indeholder litteraturhenvisninger, det nævnte minileksikon, en nyoversættelse af den pavelige bulle om kalenderreformen i 1582, ”Inter gravissimas”, forordningen om overgang i Danmark i 1700 og en liste over overgangene fra gammel til ny kalender for de forskellige lande.

Sammendrag

Forord

Denne rapport henvender sig i første række til programmører, som er interesseret i at få nogle brugbare, generelle algoritmer til dato- og tidsberegninger.

Beregning af datoer og tider dukker ustandseligt op næsten uanset hvilke problemstillinger man arbejder med, det være sig administrative eller videnskabelige. Af smertelig erfaring ved alle programmører, at det kan være meget besværligt at få dem til at virke korrekt. Mange tænker traditionelt i store tabeller, og det giver algoritmer der både fylder meget og kræver stadig vedligeholdelse.

De her viste algoritmer benytter hovedsagelig matematiske formler. Det medfører at de enkelte algoritmer ikke fylder særlig meget, og samtidig kan vedligeholdelsen holdes nede på et minimum.

Som barn samlede jeg kalenderblade og undersøgte systematikken. Astronomi, historie og tidsregning har interesseret mig lige siden, så rapporten kan anses for en samvirke af alle disse områder. Specielt blev 2 offentliggjorte algoritmer, [1] [2], banebrydende for mine overvejelser, da jeg blev voldsomt fascineret af at man kunne **regne** i datoer i stedet for at være henvist til tabeller. I den forbindelse vil jeg gerne rette en tak til professor Peter Naur, Datalogisk Institut ved Københavns Universitet, for stor inspiration, mens jeg på instituttet arbejdede med et 2.dels projekt på datalogistudiet [3].

Algoritmerne er her skrevet i et Algol/Pascal-lignende programmeringssprog, hvor definitionen af Algol-60 [4] er grundlaget for de matematiske operationer, mens der er gået let hen over en række detaljer i sproget, se nærmere i kapitel 2. Programmerne kan uden større besvær kodes i de mest kendte 3.generationssprog. For at lette forståelsen af de matematiske principper er der indsat beviser, eksempler og tabeller på passende steder. Begreber mærket med * er kort forklaret i tillæg B.

På min arbejdsplads, CSC Danmark A/S - tidligere I/S Datacentralen af 1959, har jeg hos mine nærmeste chefer mødt stor imødekommethed ved mine undersøgelser, og jeg blev i sin tid opfordret til at videreudvikle mine idéer. Det blev i løbet af ca. 3 måneder til en nogenlunde komplet "værktøjskasse" på 15-20 moduler skrevet i COBOL som kan klare de problemer, man indtil nu er stødt på.

På IBM hovedanlæg har modulerne kørt i produktion siden 1987. De eneste problemer har vist sig ved linkning af programmer på grund af forskellige versioner af COBOL. Værktøjskassen viste sig også at være et godt aktiv til at klare år2000-problemerne i Told- og Skattesystemerne.

Rapporten handler først og fremmest om den danske kalender. Fremmede landes kalendere med deres egne helligdage er kun sparsomt omtalt. Skal man opstille hele suiten af algoritmer for dem, må man gå ind i de nationale, historiske forhold. Der er dog anført enkelte af de vigtigste algoritmer for nogle landområder.

I forbindelse med arbejdet vil jeg gerne rette en varm tak til lektor cand.mag. Christian Bohr for hjælp med en nyoversættelse af den pavelige bulle fra 1582 (Tillæg C).

Lektor Nils Andersen på Datalogisk Institut ved Københavns Universitet har venligst gennemlæst rapporten og påpeget fejl og mangler, der derpå er rettet.

Til sidst en apologi: Det har ikke været min hensigt at støde nogens religiøse følelser ved at tage emnet kalender op til behandling. For nogle kan det virke chokerende at opleve en så matematisk omtale af forhold, som man anser for hellige. Jeg må derfor bede om forståelse for min matematiske nysgerrighed.

Værløse, 22. november 2004, revideret 21. marts 2005

Jens Peter Søltøft

Forord

1. Et rids af den vestlige kalenders historie

Denne fremstilling af vores nuværende kalenders historie er meget summarisk, og skal blot være en orientering og lette forståelsen af, hvordan vores kalender er blevet til.

Mere uddybende gennemgange kan man finde i litteraturen [5],[6],[7] og [8].

Vores nuværende kalender - den gregorianske* - blev 400 år den 15. oktober 1582, og det er netop tidsrummet for et helt gennemløb af den med skudår og undtagelser. Kalenderen bygger på en ældre udgave - den julianske* - og har påvirkninger med fra en række forskellige oldtidskulturer.

Den har i starten været voldsomt omstridt, bl.a. fordi den hang sammen med religiøse problemer, der var brændende for Europa i sin tid.

I dag tages det for givet at man har en rimelig præcis og velfungerende kalender, men det har bestemt ikke været nemt at komme frem hertil, og arbejdet har været en udfordring til den menneskelige snilde og opfindsomhed.

Kalendertyper

Strengt taget er opgaven med en præcis kalender nemlig matematisk uløselig! Der er ikke noget simpelt forhold mellem døgn, måned og år, og hvad værre er, deres længder ændrer sig til stadighed lidt. Men netop derfor er det lærerigt at kigge de gamle kulturer lidt i kortene med hensyn til kalenderen.

Solkalender

Ægypterne var de første, som efterlod – for eftertiden klart læselige – notater om at året ikke består af et helt antal døgn – matematisk udtrykt er der inkommensurabilitet mellem døgn og år. De lagde mest vægt på at få året bestemt så nøjagtigt som muligt på grund af Nilens oversvømmelser, som bestemte tid for såning og høst.

Så tidligt som år -3000¹ ser det ud til, at de regnede med en årslængde på 365 døgn med et skudår hvert 4. år. Fra ca. år -2000 indførte de en anden kalender på 365 døgn uden skuddage - resultatet blev altså en dobbeltkalender, men det har åbenbart ikke generet dem.

Bunden månekalender

I Mesopotamien blev der både under babylonisk og assyrisk herredømme arbejdet hårdt med præcise observationer af sol, måne og planeterne. Ugetællingen med 7 dage blev tidligt indført, og kravet om præcision og pålidelighed skyldtes hensynet til religionen med ønsker om nøjagtige forudsigelser om himmellegemernes bevægelser, der betragtedes som gudernes tegn.

De nåede også meget langt; således er det sikkert at de fandt ud af, at 235 synodiske måneder* næsten svarer til 19 tropiske år* - i dag kaldes den periode for Metons cyklus*, og den spiller stadig en vigtig rolle i vores kalender.

Da Månen indtog en central plads i religionen, kom man frem til en inddeling af året i 12 måne-måneder, hvorefter der måtte indskydes en 13. måned af og til for at få forholdet mellem år og måned til bare nogenlunde at gå op; dette kaldes en bunden månekalender*, da den er bundet til solåret. Indskudsreglerne blev efterhånden rettet ind efter Metoncyklen*; man kan regne med, at fra ca år -1000 lå deres kalender nogenlunde fast, og de havde tilmed korrektioner til den, da Metoncyklen som nævnt ikke stemmer præcist.

Kalenderen genfindes i dag i den mosaiske kalender - jøderne har netop lært den at kende i deres babyloniske fangenskabs tid.

¹ Årstal er angivet som astronomiske årstal, så år 0 astronomisk = år 1 før vores tidsregning (f.v.t.). Historikerne opererer ikke med år 0, Cæsar blev fx myrdet år 44 f.v.t = år -43 astronomisk. Fra og med år 1 efter vores tidsregning (e.v.t.) er de identiske.

1. Et rids af den vestlige kalenders historie

Fri månekalender

En fri månekalender*, hvor året består af 12 måne-måneder uden tilknytning til solåret, findes fx i den islamiske kalender.

Året er da ca. 11 dage kortere end solåret, og det har som konsekvenser bl.a., at de islamiske årstal "haler ind" på vores og de islamiske måneder flytter sig i forhold til vores solår. 33 solår svarer således til lidt mere end 34 islamiske år.

Metoncyklen – grækerne

Det ser ud til grækerne har lært at kende solåret fra ægypterne og måne-månedene fra babylonierne - hos Homer er et bundet måneår nævnt, men der har været meget vaklende indskudsregler indtil Meton omkring år -430 indførte sin cyklus som blev rettesnor i det klassiske Hellas.

Den officielle årstælling var den velkendte olympiade-tælling fra år -775.

Romerne

Ifølge sagnene opererede romerne oprindeligt med et år på 10 måneder, som ikke passede til solåret. Langsomt er de kommet frem til en bunden månekalender*, som omkring år -400 havde månederne: Martius, Aprilis, Maius, Iunius, Quintilis, Sextilis, Septembris, Octobris, Novembris, Decembris, Januarius, Februarius og indskudsmåned (Mensis intercalaris). Som det ses, var nogle måneder opkaldt efter guder, mens andre havde numre.

Romerne nåede aldrig frem til en fast indskudsregel, men plejede at skyde ekstramåned ind pludseligt, når det fx passede i præsternes politiske kram for at forlænge en konsuls embedstid. Derfor var der på Cæsars tid kommet meget forvirring i kalenderen.

Cæsars reform - den julianske kalender

I Ægypten havde Cæsar lært solåret at kende og da han også var ypperstepræst (Pontifex Maximus), gjorde han en ende på roderiet ved at indføre den julianske kalender* på 365 $\frac{1}{4}$ døgn med fast skudår hvert 4. år fra år -44. De første 20 år indsatte man fejlagtigt skudårene hver 3. år; det rettede Augustus op på ved at udelade et tilsvarende antal skuddage, og fra år +5 kom de endelig til at ligge helt fast.

Årets start

Forårsjævndøgn i kalenderen indtraf omkring 24.marts² - derved kom 1.januar til at ligge i nærheden af den korteste dag i året, og muligvis derfor blev årets start efterhånden lagt derhen. Det varede dog meget længe førend alle var enige om det. Så sent som år 1750 brugte man stadig andre starttidspunkter (juledag, 25.marts, 1.september) rundt omkring i Europa. Derfor skal man passe på med tolkningen af dokumenters datering, de kan ligge 1 år galt i forhold til vores tælling.

Romerne havde en meget besværlig måde at benævne dagene på inden for en måned – forfra til ca. midt i måneden, derefter bagfra til begyndelsen af næste måned. Først i middelalderen begyndte man at nummerere dagene fortløbende samtidig med at hver dag blev tilknyttet sine helgener.

Cæsarerne i Rom blev dyrket som guder, og derfor blev Quintilis og Sextilis omdøbt til Julius og Augustus. Tiberius fik dog som kejser nr 3 stoppet denne trafik med et spydigt: "Hvad vil I gøre, når I kommer til den 13. Cæsar?"

Kirken og kalenderen

Da Romerriget gik over til kristendommen omkring år 300, overtog kirken naturligvis den julianske kalender. Imens var ugetællingen langsomt slået igennem og søndagen blev hviledagen - måske netop for at trække skellet skarpt op overfor den jødiske tro [5], [6].

² Det præcise tidspunkt svinger med ca. 18 timer over 4 år, da årets længde som nævnt er knap 365 $\frac{1}{4}$ døgn.

1. Et rids af den vestlige kalenders historie

Kirkemødet i Nicæa – Påskedagen

Kirken havde i kalenderen et meget vigtigt problem, nemlig fastlæggelsen af den kristne påske. I kirkens barndom tog man det ikke så tungt med fastlæggelsen af højtiderne da de kristne jo alligevel forventede Verdens snarlige undergang. Men efterhånden som tiden gik, måtte man finde ud af en måde at fastlægge højtiderne på, som skal gælde for hele kristenheden. Det følte efterhånden så pinligt at måtte spørge jødiske skriftekloge til råds om en kristen højtid! Bl.a. derfor indkaldte kejser Konstantin den Store til kirkemødet i Nicæa år 325, hvor de store trosspørgsmål blev diskuteret.

På mødet blev Påskedag fastsat til første søndag efter første fuldmåne efter forårsjævndøgn. Nu er det bare svært at finde noget som har så mange uregelmæssigheder som Månens bevægelse om Jorden, så derfor enedes man - meget rimeligt - om at beregne månefaserne cyklisk* fra tabeller, dvs. gennemsnitsværdier, og efter mange besværlige eksperimenter med forskellige cyklusser, blev det til sidst Metons cyklus* der slog igennem.

Detaljerne er beskrevet mere udførligt i kapitel 7, hvor beregningen af Påskedag behandles.

For kirkefolk blev det nu muligt at opstille tabeller for Påskedagen så lang tid frem man ønskede. Denne disciplin var - med datidens ringe færdigheder i regning - ret vanskelig, da man benyttede romertallene, og de kirkefolk, som kunne mestre den, blev ligefrem kaldt **computister**, dvs. beregnere. Ved gennemgang af de forskellige kilder, [5], [8], får man næsten det indtryk, at beregningen af Påskedag var den eneste videnskabelige indsats, som oldkirken helhjertet gik ind for.

Vores årstal

På Konstantins tid anvendte man flere tællinger af årstal, nogle benyttede kejser Diokletians tronbestigelse 284, andre talte fra Roms grundlæggelse (AUC, ab urbe condita = 753 f.v.t. = -752), mens mange startede forfra ved hver ny kejsers tronbestigelse. Det gav ind mellem ret stor forvirring, og mange følte sig stødt over at tælle fra en hedensk hersker som Diokletian, der forfulgte de kristne.

I 525 foreslog munken Dionysios Exiguus at tælle fra Jesus' fødselsår - det måtte da være en værdig tidsregning. Desværre tog han fejl af placeringen - nutidige forskere er enige om han har regnet mellem 3 og 7 år galt, men hans tælling af årene er jo imens blevet godt indarbejdet.

Langsomt slog den igennem i Europa; fra ca 1400 kan man gå ud fra at den er i brug i hele kristenheden, mens man som nævnt først blev enige om årets begyndelse langt senere.

Historikerne havde så behov for at tælle år før vores tidsregning, men det har de ikke gjort helt let, se note 1.

Fejl i kalenderen hober sig op

Forårsjævndøgnet var fra -43 til 325 gledet tilbage til den 21. marts, da den julianske kalender* er lidt for lang i forhold til solåret. Jævndøgnet blev så på kirkemødet i 325 "låst fast" til 21. marts for al fremtid.

Lad gå med man ikke rettede denne forrykning; det værste var at der ikke blev foretaget noget som helst for at forhindre den i fremtiden - det ser ud som om man bare har lukket øjnene og ladet stå til. Fejlen var dog til at måle selv på den tid, ca. $1\frac{1}{4}$ min pr år \sim 1 dag på ca 130 år.

En anden fejl opstår fordi Metons cyklus* heller ikke er helt nøjagtig, den er lidt for kort i forhold til det julianske år, så Månen efterhånden kommer for tidligt i forhold til tabellerne. Fejlen er lidt over $4\frac{1}{2}$ min pr år \sim 1 dag på ca 310 år.

Disse 2 fejl hobede sig langsomt, men ubønhørligt op. Allerede den engelske munk Beda (ca 673-735), som var en af sin tids største computister, fik ved sine ret enkle observationer mistanke om afvigelser, [8], [9], og på Karl den Stores tid (år 800) var de blevet så store, at man havde travlt med bortforklaringer [5], og det blev stadig værre. Den sene middelalder havde mange gode forslag til forbedringer [6], men kirken var lammet bl.a. af indre stridigheder og dødsfald i utide af nøglepersoner, og før der skete noget var reformati- onen en kendsgerning i Nordeuropa.

Den gregorianske kalenderreform

Til sidst lykkedes det for den romersk-katolske kirke at få hold på sagerne. Ved de store modreformatorske kirkemøder (koncilier) i 1560-erne og 1580-erne blev der arbejdet med tingene [5],[6]. Det var også blevet hårdt tiltrængt: det astronomiske forårsjævndøgn var gledet tilbage til 10.-11. marts, og månefaserne var 4 dage for langt fremme i forhold til tabellerne.

Flere forslag blev overvejet, og pave Gregor 13. bestemte sig for en endelig og ret hårdhændet løsning, der var foreslået af den italienske læge Aloysius Lilius (Luigi Guilio) [5], [6].

Reformen blev beordret med den pavelige bulle ”Inter gravissimas” i februar 1582, se tillæg C.

Hovedpunkterne i reformen er:

1. En éngangsrettelse: Der rives 10 dage ud af kalenderen ved at springe direkte fra 4.oktober til 15.oktober 1582. Herved kommer det astronomiske forårsjævndøgn atter til at falde omkring 21.marts.
2. En rettelse af årets længde: Der fjernes 3 skuddage i 400 år, så kalenderårets længde bliver 365,2425 dage. Fejlen ved den nye kalender bliver da reduceret til ca 26 sekunder pr år. Hvis det tropiske år* havde konstant længde, svarer det til 1 dag pr 3300 år.
3. En rettelse til Meton cyklen: 8 gange på 2500 år rettes månetabellen, så de cyklisk* beregnede månefaser ikke igen fjerner sig fra de astronomiske*. Den nødvendige éngangsrettelse af månetabellen med 4 dage var allerede foretaget af forgængeren Pius 5. i 1568, dengang dog uden ændringer i kalenderen.

Detaljerne ved reformen kan ses i kapitel 7 og i tillæg C.

Indførelse af den gregorianske kalender

Det voldte selvsagt intet besvær at indføre kalenderen i de romersk-katolske lande:

Spanien, Portugal, de fleste italienske fyrstendømmer og dele af Polen gik over på den beordrede dato.

Frankrig nåede næsten med ved at springe fra 9.december til 20.december 1582. Katolikkerne i Holland, Schweiz, Bøhmen og Ungarn kom med inden 1587.

Tillæg E indeholder en liste, der er sammenstillet fra de forskellige kilder, men da der er uoverensstemmelser imellem dem, er hver kildes særskilte angivelser taget med.

Alle ikke-katolske stater afviste den nye kalender, da den blev opfattet som pavens, og kampen mellem reformation og modreformation rasede netop i de år - kalenderen blev én af syndebugkene i det spil. Det er betegnende for splittelsen i den videnskabelige verden, at protestanter som Tycho Brahe og Johannes Kepler gik ind for og benyttede den nye kalender i deres optegnelser, mens andre protestanter i skrifter hudflettede og smædede kalenderen. Nogle katolikker var heller ikke tilfreds med kalenderen, som de mente kunne gøres endnu bedre.

Det tyske kejserrige, der dengang var et kludetæppe af små og store stater, kom i en meget uheldig situation. Kejseren selv og de fleste katolske stater gik over i 1583-4. Resultatet blev 2 tidsregninger, og da de tyske fyrster først ved den foreløbige religionsfred i Augsburg 1555 og dernæst den Westfalske fred efter 30-års krigen i 1648 selv bestemte religionen i deres stater (”cujus regio ejus religio”), var situationen fastlåst.

Kalenderen sejrer - men langsomt

Efterhånden som den religiøse tolerance omsider vandt frem, indså man at der var store fordele ved kun at have én tidsregning, og de protestantiske stater begyndte at gå over. Nederlandene gjorde det i 1700, og det protestantiske Schweiz i 1701.

Det tyske kludetæppe fik samtidig næsten orden i sagerne ved at alle protestantiske stater (corpus evangelicorum) gik over i 1700.

I Danmark og Norge overtalte Ole Rømer den nytiltrådte konge, Frederik 4., til at gå over samtidig med de tyske stater [6]. Forordningen kan ses i tillæg D. Man udelod 11 dage og spang år 1700 fra 18.februar til

1. Et rids af den vestlige kalenders historie

1.marts, men samtidig ville man - ligesom de tyske protestantiske fyrster - beregne månefaserne efter andre tabeller. Katolikkerne skulle ikke bare triumfere! Og derved kom vores påske 2 gange til at falde 1 uge ved siden af katolikernes. 1724 fulgte kongeriget katolikkerne, mens hertugdømmerne fejrede påsken 1 uge tidligere[5], 1744 fejrede hele monarkiet påsken 1 uge før katolikkerne[10]. Først i 1776 indførtes den gregorianske påskedagsberegning sammen med de tyske protestanter.

Formodentlig på grund af de lange afstande gik Island og Færøerne først over i november 1700 [5].

I det svenske rige der dengang også omfattede Finland, ville man gå langsomt frem og begyndte med at udelade skuddagen år 1700; det var så tanken at springe 1 uge over senere på året og udelade de næste 3 skuddage, men det glemte man under den store nordiske krig, der udbrød samme år. Sverige (og Finland) fik derved sin egen tidsregning: 10 dages forskel fra gregoriansk* kalender, 1 dags forskel fra den julianske* kalender. Det varede ved indtil Karl 12. fra sit delvise fangenskab i det osmanniske rige udstedte dekret om at februar 1712 fik 2 skuddage. Så var man tilbage i den julianske kalender - og lige vidt. [5] [9].

De svenske besiddelser i Nordtyskland var allerede gået over sammen med de protestantiske stater år 1700. Først i 1753 gik Sverige og Finland over ved at springe fra 17.februar til 1.marts. Påskedagsberegningen var dog længe noget af et kaos, da den først blev gregoriansk i 1845. For Finlands vedkommende ser det ud til at den officielle tidsregning efter den russiske annekstion i 1809 gik tilbage til den julianske kalender. Først ved selvstændigheden i 1917 gik Finland endeligt over til den gregorianske kalender.

I Storbritannien og tilhørende kolonier - bl.a. Nordamerika, varede det også længe. Der var stor modstand, fordi folk bogstavelig troede, at regeringen stjal 11 dage af deres liv. Tiden var først moden i 1752, hvor der blev sprunget fra 2.september til 14.september. Der er en udførlig beskrivelse af de politiske omstændigheder i parlamentet ved reformen i [8].

De østeuropæiske lande

Modstanden mod kalenderen var større i Østeuropa, da de ortodokse kirker ikke ville vide af "den urene kalender", som ikke var godkendt på et fælles kirkemøde. Patriarken i Konstantinopel ønskede bestemt heller ikke i 1582 at lægge sig ud med den osmanniske sultan, der netop på det tidspunkt lå i krig med Spanien og de andre romersk-katolske magter. I Rusland havde Peter den Store som reform (!) indført den julianske kalender år 1700, og derved blev det indtil 1.verdenskrig.

De landområder, Rusland opslugte ved Polens delinger 1772, 1793 og 1795 og efter 1815, gik officielt tilbage til den julianske kalender [5],[11]. Zarens kancelli begyndte dog at dobbeltdatere dokumenter i 1800-tallet i korrespondance med udlandet.

Interessant er det at se de forslag, der fremkom i russiske videnskabskredse i 1800-tallet til en kalender endnu bedre end den gregorianske [5],[12], men alle disse tanker faldt til jorden under 1. verdenskrig, hvor den gregorianske kalender tilsyneladende blev indført af den kejserlige tyske besættelsesmagt i de baltiske stater i 1916-17 [11].

Lenin & co. indførte den gregorianske kalender i Rusland ved at springe 13 dage i begyndelsen af 1918. Oktoberrevolutionen i 1917 fandt derfor sted 7.november efter vores kalender.

De græsk-ortodokse balkanlande antog under og efter 1.verdenskrig en egen forbedret kalender, som indtil år 2800 vil være den samme som den gregorianske - der er lidt andre skudårsregler [5]. Den forbedrede kalender blev taget i brug mellem 1916 og 1923, men kirken regner stadig internt med den gamle kalender. Derfor falder vores Hellig 3 Konger, 6.januar, Julens Ende, i øjeblikket sammen med den ortodokse Jul, da der mellem 1900 og 2100 er 13 dages forskel mellem kalenderne.

Den franske revolutionskalender

Det eneste alvorlige forsøg, som i nyere tid har været gjort for at få en anden kalender, var indførelsen af revolutionskalenderen i Frankrig fra oktober 1793.

Starttidspunktet blev lagt til efterårsjævndøgn, 22.september 1792, hvor man indførte republikken. Året blev inddelt i 12 måneder à 3 dekader à 10 dage, og samtidig blev dagen inddelt i 10 timer à 100 minutter.

1. Et rids af den vestlige kalenders historie

Månederne fik nye strengt rationelle navne³, og de 5 overskydende dage blev fejret som festdage, viet til dyden, geniet, arbejdet, overbevisningen og lønnen. Hvert 4. år havde så en 6. overskudds dag, revolutionsdagen, hvor alle skulle aflægge fornyet ed på at leve og dø i frihed. I dag kan man trække på smilebåndet af det, men revolutionsfolkene mente det skam gravalvorligt; hvis man tvivlede, kunne det let koste hovedet!

Den kalender var dog i praksis for stærk kost for folk - de ville have hviledag hver 7. dag og ikke hver 10; dekaderne gled hurtigt ud igen, og det franske akademi udarbejdede en sprænglærd afhandling i 1804, hvor man påpegede, at ingen andre lande ville antage revolutionskalenderen. Den blev endelig afskaffet af Napoleon 31. december 1805. Under pariserkommunen i 1871 ville man genoplive den, så der kom enkelte dekretter med datering i revolutionskalenderen [5].

Tankevækkende er det, at en tilsvarende institution fra den franske revolution, nemlig metersystemet, overlevede - dog først efter at være afskaffet ligesom kalenderen.

Fremtiden

Den gregorianske kalender har sine ulemper - datoerne falder ikke hvert år på de samme ugedage og påsken flytter rundt med de forskydelige højtider. Månederne er ikke lige lange og reglerne for skuddage er ret indviklede.

Hvis man vil have en kalender, hvor en dato har samme ugedag år efter år, er man nødt til at afbryde ugerne, da 7 hverken går op i 365 eller 366, og problemerne må klares ved indskud af ekstra 'ugeløse' dage.

Generelt skræmmer menneskehedens erfaringer dog: Det har nu taget noget over 400 år at blive enige om **én kalender**, der bruges internationalt. FN har haft sagen op og vende i 1951 og senere [6], men den har åbenbart samlet støv dér. Der har været opstillet 2 skemaer [13] til en ny forbedret kalender:

- **Fast Kalender** (fixed calendar) består af 13 måneder à 28 dage. Månednavnene beholdes, og mellem Juni og Juli kommer en ny måned, kaldet Sol. Hver måned begynder på en søndag og slutter med en lørdag. Hvert år indsættes der en navnløs ekstra dag efter 28. december, og i skudår også en navnløs skuddag efter 28. juni. Den største ulempe er nok at man ikke mere kan operere med kvartaler eller halvår.
- **Verdens Kalender** (world calendar) består af 12 måneder, grupperet i 4 kvartaler på 91 dage. Månedslængderne i et kvartal er 31, 30 og 30 dage. Hvert kvartal begynder med en søndag. På samme måde som i den faste kalender indsættes der hvert år en navnløs ekstra dag efter 30. december, og i skudår også en skuddag efter 30. juni. Fordelen er at hvert kvartal består af 13 hele uger, og ugedagsmønsteret gentager sig i hvert kvartal.

Den gregorianske kalender har også sine fejl. Forårsvævsdøgnnet vil stadig bevæge sig tilbage - omkring år 10000 vil det være 15. marts; det skyldes dog hovedsagelig, at det tropiske års* længde i øjeblikket aftager lidt. Et eller andet sted mellem år 3200 og 3600 bør man derfor udelade en første skuddag.

Beregningen af månefaser vil give en fejl på 1 dag på ca 22000 år.

Fremtidige rettelser i den gregorianske kalender vil ikke volde de store problemer, da den allerede nu har indbygget de 2 korrektioner (pkt. 2 og 3 i reformen).

Man kan blot internationalt – hvad der jo kan volde store kvaler! - vedtage at foretage ekstra korrektioner efter behov, så vil forårsvævsdøgn, månefaser, ugedage og beregningen af påsken automatisk blive rettet.

Status i dag

Kalenderen har efter 420 års brug vist sin levedygtighed, men der er et stykke vej igen, førend den har været i brug lige så længe som den julianske. Først omkring år 3500 har den nået sin modne alder!

³ Navnene er: Vendémiaire, Brumaire, Frimaire, Nivôse, Pluviose, Ventôse, Germinal, Floréal, Prairial, Messidor, Thermidor, Fructidor. Man kan således se, at hver årstid afspejler sig i navnene.

2. Kontrol af en dato

Kort introduktion til algoritmerne

Algoritmerne er skrevet i et Algol/Pascal-lignende programmeringssprog, hvor definitionen af Algol-60 [4] er grundlaget for de matematiske operationer, mens en række detaljer i denne fremstilling er udeladt, fx problemstillingen call-by-value overfor call-by-name og evt. overflødige ; i forbindelse med **end**.

Der er 3 hovedtyper af variable, nemlig **boolean**, **integer** og **real**. Arrayelementer er angivet i [], mens kald af procedurer er angivet i (). I reelle tal anvendes decimalkomma.

Relationerne 'større end eller lig', 'mindre end eller lig' og 'forskellig fra' angives med hhv \geq , \leq og \neq .

Sammensatte betingelser angives med **or**, **and** og **not**

Kommentarer er angivet i /* - - - */.

Der opereres med 2 typer løkker: **for – step – until – do** og **while – do**.

Alle algoritmer er skrevet som fritstående procedurer, der deles i 2 kategorier:

- type-procedurer, der returnerer én værdi (boolean, integer, real). Den tilhørende parameterliste rummer kun inddata.
- andre procedurer, der returnerer værdier gennem parameterlisten.

Med **return** angives, hvad begge kategorier returnerer.

En procedure forlades med **quit**, hvis det er så alvorlige fejl, at videreberegning er umulig.

I den situation kan man supplere med en returnerende fejlkode.

Operatortegn

/ betegner den almindelige division.

I Algol 60-rapporten [4] anvendes det amerikanske divisionstegn \div til at betegne heltalsdivision. For at undgå forvekslinger med andre operatører benyttes i rapporten konsekvent dobbeltsymbolet // i stedet for.

Modulus (divisionsrest) angives med **mod**. De præcise definitioner kan ses i kapitel 3.

Hvornår skal en dato kontrolleres

Før man kaster sig ud i kalenderberegninger, bør man være i stand til rimelig let at kunne undersøge **om** en forelagt dato overhovedet eksisterer. Men det er lige så vigtigt at gøre sig klart, **hvornår** det er nødvendigt at foretage kontrollen.

I it-systemer kommer datoer (år-måned-dag eller år-dag) fra disse 4 kilder (datafødsel/datafangst) - og netop kun fra disse:

- **Lagret**
En dato hentes frem til brug fra et register, en fil eller en database.
Den skal ikke kontrolleres - det skal den sandelig have været, længe inden den blev gemt!
- **Indtastet/indlæst**
Sådan en dato er ny og rå. Den skal kontrolleres nu, før den kan indgå i beregninger.
- **Beregnet**
En dato kommer som et resultat af en beregning.
Den skal ikke kontrolleres - ellers kan man jo ikke stole på den beregning!
- **Hentet fra maskinen**
En dato kommer fra kald af en systemrutine på datamaskinen.
Den skal ikke kontrolleres - ellers er der noget galt med installationen!

2. Kontrol af en dato

Alle kontroller - bortset fra dem på netop indtastede/indlæste datoer - er ofte det rene tidsspilde! Bare tænk på, om man i et almindeligt programmeringssprog før hver eneste addition **selv** skulle kontrollere, om de indgående operander nu også er regulære tal.

År-2000-problemet opstod netop fordi man **ikke** opererede med korrekte datoer, hverken som indtastede eller lagrede. Derfor kunne man ikke stole på dem og var nødt til de enormt kostbare undersøgelser af dato- og årstalsangivelser alle vegne i systemerne – uanset hvilken kilde de oprindeligt kom fra.

Er man derimod sikker, så kan man nøjes med at kontrollere indtastede/indlæste datoer, og kun dem!

2 algoritmer: Skudårsundersøgelse og datokontrol

Selve kontrollen foretages mest effektivt ved direkte at benytte kalenderens egenskaber. Det viser sig, at man med fordel kan splitte kontrollen op i 2 algoritmer:

- én til at finde ud af om et år er et almindeligt år, skudår, eller et kortere overgangså mellem de forskellige kalendere.
- én til selve kontrollen af datoer.

Skudårsundersøgelsen kan nemlig også indgå som led i nogle af beregningsrutinerne.

De forskellige kalendere

Der kommer et problem, når en dato skal kontrolleres: Overgangen fra juliansk* til gregoriansk* kalender fandt som nævnt i kapitel 1 ikke sted på samme tidspunkt i de forskellige lande i Europa, og derfor vises her algoritmerne for Danmark og Norge.

Tillæg E indeholder en oversigt, kapitel 11 kodeeksempler for nogle andre lande.

Algoritme 1: ÅRTYPE, Danmark og Norge

 /* Å R T Y P E */

integer procedure årtype(år);

integer år;

/* Bestemmer årets længde, gælder for dansk-norsk kalender 600 – 3199,

årtype:

- 0 fejl, året er udenfor intervallet 600 – 3199
- 1 almindeligt år (365 dage)
- 2 skudår (366 dage)
- 3 overgangsåret 1700 (355 dage) */

begin

if år < 600 **or** år > 3199

then årtype := 0

/* fejl i årstal */

else begin

if år > 1700

then begin

/* gregoriansk kalender */

if år **mod** 4 \neq 0

then årtype := 1

/* almindeligt år */

else begin

if år **mod** 100 \neq 0

then årtype := 2

/* almindeligt skudår */

else begin

if år **mod** 400 \neq 0

then årtype := 1;

/* 100-år, almindeligt år */

else årtype := 2

/* 400-år, skudår */

end

end

end

else begin

if år < 1700

then begin

/* juliansk kalender */

if år **mod** 4 \neq 0

then årtype := 1;

/* almindeligt år */

else årtype := 2

/* skudår */

end

else årtype := 3

/* år 1700, overgangsåret */

end

end;

return årtype;

end ÅRTYPE

2. Kontrol af en dato

Algoritme 2: VALIDATO, Danmark og Norge

```
/* V A L I D A T O */
```

```
integer procedure validato(år,md,dg);
```

```
integer år,md,dg;
```

```
/* Kontrollerer, at en dato er valid (gyldig), gælder for dansk-norsk kalender 600-3199.
```

```
validato:
```

```
0 ok
```

```
1 fejl i dg
```

```
2 fejl i md
```

```
4 fejl i år.
```

```
Summer kan forekomme ved kombinerede fejl */
```

```
begin
```

```
validato := 0;
```

```
/*kontrol af dag, fejl med feb 29 fanges til sidst */
```

```
if dg < 1 or dg > 31
```

```
then validato := 1
```

```
/* fejl i dag */
```

```
else begin
```

```
  if md = 4 or md = 6 or md = 9 or md = 11
```

```
  then begin
```

```
    if dg > 30
```

```
    then validato := 3
```

```
/* fejl i måned - dag */
```

```
  end
```

```
  else begin
```

```
    if md = 2 and dg > 29
```

```
    then validato := 3
```

```
/* fejl i måned - dag */
```

```
  end
```

```
end;
```

```
/* kontrol af måned, validato = 3 her umuliggør validato = 2 nedenfor */
```

```
if md < 1 or md > 12
```

```
then validato := validato + 2;
```

```
/* fejl i måned */
```

```
/* kontrol af år - og dermed interval */
```

```
if årtype(år) = 0
```

```
then validato := validato + 4;
```

```
/* fejl i år */
```

```
/* kontrol af feb 29 */
```

```
if årtype(år) = 1 and md = 2 and dg = 29
```

```
then validato := 7
```

```
/* fejl i år - måned - dag */
```

```
else begin
```

```
/* kontrol af overgangsåret 1700 */
```

```
  if årtype(år) = 3 and md = 2 and dg > 18
```

```
  then validato := 7
```

```
/* fejl i år - måned - dag */
```

```
end;
```

```
return validato;
```

```
end VALIDATO
```

3. Grundformler til kalenderberegninger

Heltalsregning

Heltalsregning består af de 4 regningsarter, hvor den almindelige division dog normalt er erstattet af **heltalsdivision** (kvotient) og **divisionsrest** (modulus).

Multiplikationstegnet * vil kun undtagelsesvis blive anvendt, ellers er det underforstået.

I alle divisioner og heltalsdivisioner er det forudsat, at nævneren ikke er 0.

Der er et grundlæggende problem ved heltalsregningen, da samme operator defineres **forskelligt** i de forskellige kilder [18], [14], og derfor kommer der en sammenligning og diskussion til sidst i kapitlet!

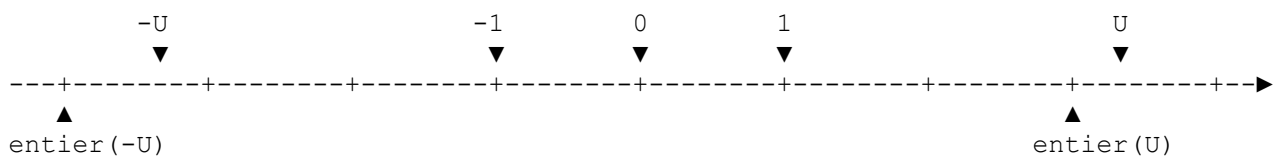
Generelle definitioner

Definitionerne af **entier**, **sign** og **abs** er fælles i de benyttede kilder:

Entier (heltalsdel)

Heltalsdelen (også kaldet "gulvet") af et udtryk er defineret ved:

$$\text{entier}(U) = \lfloor U \rfloor = \text{det største heltal } \leq U \quad (1)$$



Eksempler:

$$\begin{array}{lll} \text{entier}(2) = 2 & \text{entier}(1,3) = 1 & \text{entier}(3,7) = 3 \\ \text{entier}(0) = 0 & & \\ \text{entier}(-2) = -2 & \text{entier}(-1,3) = -2 & \text{entier}(-3,7) = -4 \end{array}$$

Fortegn (sign)

Fortegnet for et udtryk er defineret ved:

$$\text{sign}(U) = \begin{cases} \text{for } U > 0 : & +1 \\ \text{for } U = 0 : & 0 \\ \text{for } U < 0 : & -1 \end{cases} \quad (2)$$

Eksempler:

$$\begin{array}{lll} \text{sign}(2) = 1 & \text{sign}(1,3) = 1 & \text{sign}(3,7) = 1 \\ \text{sign}(0) = 0 & & \\ \text{sign}(-2) = -1 & \text{sign}(-1,3) = -1 & \text{sign}(-3,7) = -1 \end{array}$$

3. Grundformler til kalenderberegninger

Absolut værdi (numerisk værdi)

Den absolutte værdi (også kaldet den numeriske værdi) for et udtryk er defineret ved:

$$\text{abs}(U) = \begin{cases} \text{for } U \geq 0 : & +U \\ \text{for } U < 0 : & -U \end{cases} \quad (3)$$

Eksempler:

$$\begin{array}{lll} \text{abs}(2) = 2 & \text{abs}(1,3) = 1,3 & \text{abs}(3,7) = 3,7 \\ \text{abs}(0) = 0 & & \\ \text{abs}(-2) = 2 & \text{abs}(-1,3) = 1,3 & \text{abs}(-3,7) = 3,7 \end{array}$$

Heltalsdivision

Med denne definition begynder problemerne !
Heltalsdivisionen er i [4] og [14] defineret ved:

$$a \div b = \text{sign}(a/b) * \text{entier}(\text{abs}(a/b)) \quad (4a)$$

/ betegner den almindelige division. Prioriteten for \div er den samme som for $*$ og $/$

Men i [18] er heltalsdivisionen defineret ved:

$$a \div b = \lfloor a / b \rfloor \quad (4b)$$

Modulus (divisionsrest)

Som supplement til de 4 regningsarter er det bekvemt at indføre modulusbegrebet. Modulus er dog ikke strengt nødvendig at have, da den kan udtrykkes ved de andre regningsarter.

Igen er definitionerne forskellige i [14] og [18] !

IBMs C-lister (command-listers) [15] og definitionen i IBM Visual Age for Java for Windows 3.5/4.0 er begge implementeret som [14]. Det er konstateret ved simpel afprøvning med rod i almindelig nysgerrighed !

Modulus er i [14] defineret ved:

$$a \bmod b = a - a \div b * b \quad (5a)$$

mod har samme prioritet som \div og $/$ og $*$

I de matematiske udledninger er der dog mange steder sat ekstra parenteser omkring modulusudtryk for forståelsens skyld.

Mens modulus i [18] er defineret ved:

$$a \bmod b = a - b * \lfloor a/b \rfloor \quad (5b)$$

3. Grundformler til kalenderberegninger

Sammenligning af de 2 sæt definitioner af heltalsdivision og modulus

Sammenligner man de 2 sæt af definitioner – enten ved direkte omskrivning af formlerne, eller som her gennem en tabel - kommer man frem til:

		Definition efter Algol60 [4]		Definition efter Conc.mat [18]				Definition efter Algol60 [4]		Definition efter Conc.mat [18]		
		entier		a mod		entier		a mod		a mod		
a	a/3	(a/3)	a÷3	3	a÷3	3	a/(-3)	(a/(-3))	a÷(-3)	(-3)	a÷(-3)	(-3)
-9	-3	-3	-3	0	-3	0	3	3	3	0	3	0
-8	-2,7	-3	-2	-2	-3	1	2,7	2	2	-2	2	-2
-7	-2,3	-3	-2	-1	-3	2	2,3	2	2	-1	2	-1
-6	-2	-2	-2	0	-2	0	2	2	2	0	2	0
-5	-1,7	-2	-1	-2	-2	1	1,7	1	1	-2	1	-2
-4	-1,3	-2	-1	-1	-2	2	1,3	1	1	-1	1	-1
-3	-1	-1	-1	0	-1	0	1	1	1	0	1	0
-2	-0,7	-1	0	-2	-1	1	0,7	0	0	-2	0	-2
-1	-0,3	-1	0	-1	-1	2	0,3	0	0	-1	0	-1
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0,3	0	0	1	0	1	-0,3	-1	0	1	-1	-2
2	0,7	0	0	2	0	2	-0,7	-1	0	2	-1	-1
3	1	1	1	0	1	0	-1	-1	-1	0	-1	0
4	1,3	1	1	1	1	1	-1,3	-2	-1	1	-2	-2
5	1,7	1	1	2	1	2	-1,7	-2	-1	2	-2	-1
6	2	2	2	0	2	0	-2	-2	-2	0	-2	0
7	2,3	2	2	1	2	1	-2,3	-3	-2	1	-3	-2
8	2,7	2	2	2	2	2	-2,7	-3	-2	2	-3	-1
9	3	3	3	0	3	0	-3	-3	-3	0	-3	0

- Hvis a og b har **samme fortegn**, giver begge definitioner **samme resultater** !

Og da der i kalenderberegningerne overalt opereres med **ikke-negative** heltal, bevirker det at man frit kan operere med et modulusbegreb, som ”dækker” begge definitionerne.

I resten af rapporten vil **heltalsdivision** og **modulus** derfor stå for begge definitioner, hvor det er forudsat, at alle mellemresultater er ikke-negative.

Som operator for heltalsdivisionen benytter Algol 60 som nævnt i kapitel 2 det amerikanske divisionstegn ÷. I denne rapport er operatoren overalt erstattet med //.

Formel fra definitionerne

Af definitionen (5a) eller (5b) får man ved almindelig division med b, så længe der kun opereres med ikke-negative tal:

$$(a \text{ mod } b)/b = a/b - a//b$$

og overflytning:

$$a//b = (a - a \text{ mod } b)/b \tag{10}$$

3. Grundformler til kalenderberegninger

Specielle forhold i Ada

Programmeringssproget Ada tilbyder begge definitioner af modulus [17]:

Definitionstype	Terminologi i Ada
Algol 60 [4], RC Algol 6 [14]	rem
Concrete mathematics [18]	mod

4. Interkalation - definition og eksempler

Egenskaber ved kalenderberegninger

Kalenderberegninger har 4 egenskaber, der adskiller dem fra sædvanlige beregninger:

1. Det er hovedsagelig **regning med heltal**, (fx et antal dage). Eventuelle brøkdeler af de ønskede størrelser kan altid beregnes efter man har fastlagt foreløbige værdier i hele tal.
2. Beregningerne har **cyklisk karakter**, dvs. de gentages inden for hver cyklus.
3. Der er **ikke et heltalligt forhold** mellem forskellige perioder i en cyklus, fx vil et bestemt antal dage ikke svare til et fast antal måneder.
4. De enkelte perioder er **ikke altid lige lange**, fx er et skudår længere end de almindelige år.

Enhed, periode, cyklus

I denne rapport anvendes betegnelserne enhed, periode og cyklus således:

- En **enhed** er den mindste tidsangivelse. Alle enheder har samme længde.
- En **periode** består af et varierende antal enheder.
- En **cyklus** består af et antal perioder, og har et fast antal enheder.

Eksempel 4.1

Enhed = 1 døgn, periode = 1 måned med mellem 28 og 31 enheder, cyklus = 1 år (= 365 døgn).
Perioderne svinger i længde, mens cyklus har fast længde.

Interkalation

Ved kalenderberegninger har man brug for at foretage interkalationer*. En interkalation (indsrud i en kalender, engelsk: intercalation) kan indsætte ekstra enheder (fx dage), i perioder (fx måneder) inden for en cyklus (fx år).

Regelmæssig interkalation

Interkalationen kaldes **regelmæssig**, når de forskellige længder repeteres (gentages) indenfor hver cyklus.

Eksempel 4.2

Periode = 1 år, cyklus = 4 år, indsættelse af et skudår på 366 dage mellem 3 år på 365 dage.
Denne interkalation er regelmæssig, da den repeteres: 365, 365, 365, 366; 365, 365, 365, 366; 365 ...

Eksempel 4.3

Periode = 1 måned, cyklus = 4 år, indsættelse af måneder på 28, 29, 30 og 31 dage imellem hinanden.
Denne interkalation er ikke regelmæssig, da længden af februar måned vil ændre sig inden for en cyklus.

4. Interkalation – definition og eksempler

Jævn interkalation

Hvis en regelmæssig interkalation yderligere:

- kun har en forskel i periodernes længde på 1,
- og de uens periodelængder er fordelt jævnt på en sådan måde, at de **alle** kan beregnes som differenserne mellem 2 følgende perioder **med heltalsdivision** – dvs. med nedrunding af en almindelig division, kaldes den en **jævn interkalation**.

En jævn interkalation vil fremkomme, når man beregner de enkelte perioders start med heltalsdivisionen

$$(A \cdot p) // Q$$

hvor cyklussens faste længde er på A enheder

p er periodens nummer i cyklussen

Q er cyklussens længde i perioder

og hvor de enkelte perioder har en længde på enten $A // Q$ eller $A // Q + 1$ enheder

Eksempel 4.2 er en sådan, da periodelængden kun skifter med 1, og man kan beregne længderne af perioderne ved differensen mellem heltalskvotienterne af typen $(1461 \cdot p) // 4$, hvor $p = 1, 2, 3, 4, \dots$

Eksempel 4.4 og 4.5

En regelmæssig interkalation med repetition 28, 28, 28, 27, 27; 28, ... er ikke jævn, da den ikke kan udtrykkes ved en sådan heltalsdivision, mens repetitionen 27, 28, 27, 28, 28; 27, ... vil være jævn.

Formler for jævne interkalationer

Det er muligt at opstille 3 sammenhørende formler for jævne interkalationer, hvor man omregner et totalt antal enheder i en cyklus til det tilsvarende periodenummer og enhedsnummer i perioden $u \blacktriangleright (p, d)$ og omvendt $(p, d) \blacktriangleright u$:

$$u = (A \cdot p + B) // Q + d \tag{30}$$

$$p = (Q \cdot u - 1 - B) // A \tag{31}$$

$$d = ((Q \cdot u - 1 - B) \bmod A) // Q + 1 \tag{32}$$

Alle indgående tal er heltal,

$$A > 0, Q > 1, p \geq 0, u \geq 0, d \geq 0, A \cdot p + B \geq 0, Q \cdot u - 1 - B \geq 0$$

Konstanterne er:

A cyklussens længde, målt i antal enheder

Q cyklussens længde, målt i antal perioder

B fastlægger, hvor repetitionen af perioderne skal ligge.

Værdien af B benyttes også til at bestemme det lavest tilladte nummer for perioderne, dvs. nummereringen af dem.

Da det forudsættes, at der opereres på ikke-negative tal, skal produktet $A \cdot p$ have en modsvarende stor positiv værdi, hvis B har en relativ stor negativ værdi, før formlerne (30), (31) og (32) gælder. Den første brugbare værdi af p bliver da det lavest mulige nummer for perioderne.

Idéen i beviset er efter indsættelse at køre formlen for heltalsdivision ”baglæns”, så der kommer almindelige divisionstegn frem, og derved kan udtrykket reduceres. Sidste trin er at vurdere og vise, at udtrykket $x \bmod Q + y \bmod Q$ kan reduceres til $(x + y) \bmod Q$.

4. Interkalation – definition og eksempler

Bevis for formlerne

Ved indsættelse af (31) og (32) i højresiden af (30):

$$\begin{aligned} T &= (Ap + B)//Q + d \\ &= (A((Qu - 1 - B)//A) + B)//Q + ((Qu - 1 - B) \bmod A)//Q + 1 \\ &= (A((Qu - 1 - B)//A) + B)//Q + (Qu - 1 - B - Ap)//Q + 1 \end{aligned} \quad (33)$$

Lad nu:

$$K = Qu - 1 - B \quad (34)$$

Så vil (33) blive til:

$$T = (A(K//A) + B)//Q + (K - Ap)//Q + 1$$

Og ved fornyet indsættelse af (31) og omformning ved (34):

$$T = (A(K//A) + B)//Q + (K - A(K//A))//Q + 1 \quad (35)$$

Indsættes (10) $[a//b = (a - a \bmod b)/b]$ i (35), får man så:

$$\begin{aligned} T &= (A((K - K \bmod A)/A) + B)//Q + (K - A((K - K \bmod A)/A))//Q + 1 \\ &= (K - K \bmod A + B)//Q + (K - K + K \bmod A)//Q + 1 \end{aligned} \quad (36)$$

Indsættes så (34) udenfor udtrykkene med modulus:

$$\begin{aligned} T &= (Qu - 1 - B - K \bmod A + B)//Q + (K \bmod A)//Q + 1 \\ &= (Qu - 1 - K \bmod A)//Q + (K \bmod A)//Q + 1 \end{aligned} \quad (37)$$

Indsættes så (10) $[a//b = (a - a \bmod b)/b]$ i (37):

$$\begin{aligned} T &= (Qu - 1 - K \bmod A - (Qu - 1 - K \bmod A) \bmod Q)/Q \\ &\quad + (K \bmod A - (K \bmod A) \bmod Q)/Q + 1 \\ &= (Qu - 1 - K \bmod A - (Qu - 1 - K \bmod A) \bmod Q + K \bmod A - (K \bmod A) \bmod Q)/Q + 1 \\ &= (Qu - 1 - (Qu - 1 - K \bmod A) \bmod Q - (K \bmod A) \bmod Q)/Q + 1 \end{aligned} \quad (38)$$

4. Interkalation – definition og eksempler

For at komme videre, skal man vurdere udtrykket:

$$x \bmod Q + y \bmod Q \tag{39}$$

hvor:

$$\begin{aligned} x &= Qu - 1 - K \bmod A \\ &= Qu - 1 - (Qu - 1 - B) \bmod A \end{aligned} \tag{39a}$$

$$\begin{aligned} y &= K \bmod A \\ &= (Qu - 1 - B) \bmod A \end{aligned} \tag{39b}$$

$$Q > 1, A > 0, u \geq 0$$

Når u begynder at vokse, så vil både (39a) og (39b) fra et vist trin være positive, hvorved der ingen problemer er ved at benytte heltalsdivision og modulus.

Af (39a) og (39b) kan man direkte beregne:

$$\begin{aligned} (x + y) \bmod Q &= (Qu - 1 - K \bmod A + K \bmod A) \bmod Q \\ &= (Qu - 1) \bmod Q \\ &= Q - 1 \\ &< Q \end{aligned} \tag{40}$$

Men når (40) gælder, og alle 3 størrelser er positive, så:

$$x \bmod Q + y \bmod Q = (x + y) \bmod Q \tag{41}$$

Så kan (38) reduceres til :

$$\begin{aligned} T &= (Qu - 1 - (Qu - 1 - K \bmod A + K \bmod A) \bmod Q)/Q + 1 \\ &= (Qu - 1 - (Qu - 1) \bmod Q)/Q + 1 \end{aligned} \tag{42}$$

Og ved indsættelse af (40):

$$\begin{aligned} T &= (Qu - 1 - (Q - 1))/Q + 1 \\ &= u \end{aligned}$$

Hvilket var det, der skulle vises.

Formlerne for jævne interkalationer gælder altså med samme forbehold som heltalsdivision og modulus: Mellemlregninger, der giver negative resultater, kan ikke anvendes.

4. Interkalation – definition og eksempler

Eksempler på interkalation

Eksempel 4.6

Et minimaleksempel.

Cyklus = 3 perioder = 8 enheder, repeteret således: 3, 3, 2; 3, 3, 2; 3, ...

Periode = 2 eller 3 enheder

$A = 8$, $B = -6$, $Q = 3$, interkalationsformlerne (30), (31) og (32) bliver da til:

$$u = (8p - 6) // 3 + d$$

$$p = (3u + 5) // 8$$

$$d = ((3u + 5) \bmod 8) // 3 + 1$$

Tabel 4.6																			
Jævn interkalation - minimaleksempel																			
(p, d) til u																			
$u = (8p - 6) // 3 + d$																			
p \ d	cyklus 1			cyklus 2			cyklus 3			cyklus 4			osv ...						
	1	2	3	4	5	6	7	8	9	10	11	12							
1	1	4	7	9	12	15	17	20	23	25	28	31							
2	2	5	8	10	13	16	18	21	24	26	29	32							
3	3	6		11	14		19	22		27	30								
u til (p, d)																			
$p = (3u + 5) // 8, \quad d = ((3u + 5) \bmod 8) // 3 + 1$																			
u	p	d					u	p	d										
1	1	1	periode 1	cyklus 1			17	7	1	periode 7	cyklus 3								
2	1	2					18	7	2										
3	1	3					19	7	3										
4	2	1	periode 2				cyklus 2			20				8	1	periode 8	cyklus 4		
5	2	2								21				8	2				
6	2	3								22				8	3				
7	3	1	periode 3	cyklus 2			23	9	1	periode 9	cyklus 4								
8	3	2					24	9	2										
9	4	1	periode 4	cyklus 2			25	10	1	periode 10	cyklus 4								
10	4	2					26	10	2										
11	4	3					27	10	3										
12	5	1	periode 5	cyklus 2			28	11	1	periode 11	cyklus 4								
13	5	2					29	11	2										
14	5	3					30	11	3										
15	6	1	periode 6	cyklus 2			31	12	1	periode 12	cyklus 4								
16	6	2					32	12	2										

4. Interkalation – definition og eksempler

Eksempel 4.7

Den jævne interkalation år-dage i en 4-års cyklus.

Det er den julianske kalender* med den permanente 4-årige skudårsperiode.

Cyklus = 4 år = 1461 døgn, repeteret således: 366, 365, 365, 365; 366, ...

Periode = 1 år = 366 eller 365 døgn.

A = 1461, B = 3, Q = 4, interkalationsformlerne bliver da:

$$u = (1461p + 3) // 4 + d$$

$$p = (4u - 4) // 1461$$

$$d = ((4u - 4) \bmod 1461) // 4 + 1$$

Tabel 4.7														
Jævn interkalation - 4 årig cyklus i den Julianske kalender														
(p, d) til u														
u = (1461p + 3) // 4 + d														
p \ d	cyklus 1				cyklus 2				cyklus 3				cyklus 4 ▶	
	periode				periode				periode					
	0	1	2	3	4	5	6	7	8	9	10	11	12	
1	1	367	732	1097	1462	1828	2193	2558	2923	3289	3654	4019	4384	
2	2	368	733	1098	1463	1829	2194	2559	2924	3290	3655	4020	4385	
3	3	369	734	1099	1464	1830	2195	2560	2925	3291	3656	4021	4386	
.....	
364	364	730	1095	1460	1825	2191	2556	2921	3286	3652	4017	4382	4747	
365	365	731	1096	1461	1826	2192	2557	2922	3287	3653	4018	4383	4748	
366	366				1827				3288				4749	
u til (p, d)														
p = (4u - 4) // 1461, d = ((4u - 4) mod 1461) // 4 + 1														
u	p	d					u	p	d					
1	0	1	periode 0	cyklus 1 ▼	1095	2	364	▲	periode 2	▲	cyklus 1			
2	0	2												1096
.....				1097	3	1						
365	0	365							1098					3
366	0	366									
367	1	1	periode 1		1461	3	365	▼	periode 3	▼	cyklus 2			
368	1	2												1462
.....				1463	4							
730	1	364											
731	1	365				1826	4							365
732	2	1	periode 2	1827		4	366	▼	periode 4	▼	cyklus 2			
733	2	2												1828
.....				1829	5							
			1829						5					2

Flere interkalationer ovenover hinanden

Med lidt behændighed kan man operere med flere interkalationer ovenpå hinanden.

Med eksempel 4.7 som udgangspunkt man fx lægge en ny interkalation med en 100-års eller en 400-års cyklus ovenpå 4-års cyklussen.

Ved konstruktionen af nogle af kalenderalgoritmerne i de kommende kapitler vil det princip blive brugt igen og igen.

4. Interkalation – definition og eksempler

Eksempel 4.8

Jævn (men fiktiv) interkalation måneder-dage i en års cyklus.

Hvis februar havde 30 dage, ville det være muligt at skabe en jævn interkalation af månederne over et år.

Nu får den i første omgang råt og brutalt tildelt 30 dage for at få en jævn interkalation. Derpå indføres så en korrektion i eksempel 4.9, der vil få den uregelmæssige (og dermed ikke jævne) interkalation til at fungere.

Der er en fleksibilitet ved den jævne interkalation: Man kan klare sig med en cyklus, som er **mindre** end den angivne, hvis en mindre cyklus repeterer mønstret for periodernes varierende længder korrekt. Den mindre cyklus behøver heller ikke at gå op i den givne.

Af pladshensyn angives kun formlerne for den sidste opsætning.

Den givne cyklus fra starten af = 1 ”år” = 12 måneder = 367 dage

Periode = 1 måned (7 à 31 dage + 5 à 30 dage)

Enhed = 1 dag

Repetitionsmønster = 31, 30, 31, 30, 31, 31, 30, 31, 30, 31, 30, 31, men det kan brækkes over i 2 dele:

31, 30, 31, 30, 31, 30, 31; 31, 30, 31, 30, 31, dvs.

aktuel mindre cyklus = 7 måneder = 214 dage (4 måneder à 31 dage + 3 à 30 dage).

For denne interkalation skal der være disse værdier:

A = 214, B = -211, Q = 7, og interkalationsformlerne bliver:

$$u = (214p - 211) // 7 + d$$

$$p = (7u + 210) // 214$$

$$d = ((7u + 210) \bmod 214) // 7 + 1$$

Tabel 4.8													
Jævn interkalation - 1 fiktivt år													
(p, d) til u													
u = (214p - 211) // 7 + d													
p \ d	cyklus 1, periode							cyklus 2, periode					
	1	2	3	4	5	6	7	8	9	10	11	12	
1	1	32	62	93	123	154	184	215	246	276	307	337	resten
2	2	33	63	94	124	155	185	216	247	277	308	338	bruges
.....	ikke
29	29	60	90	121	151	182	212	243	274	304	335	365	
30	30	61	91	122	152	183	213	244	275	305	336	366	
31	31		92		153		214	245		306		367	

u til (p, d)

$$p = (7u + 210) // 214, \quad d = ((7u + 210) \bmod 214) // 7 + 1$$

Det er lige før, det er en brugbar kalender, og med en mindre korrektion kan man opstille den første algoritme, da de ønskede værdier kun afviger fra eksempel 4.8 med en konstant værdi fra marts og året ud. For januar og februar er værdierne i orden.

Dvs. man kan benytte eksempel 4.8, hvis der indføres en rettelse for marts måned og frem, se eksempel 4.9.

4. Interkalation – definition og eksempler

Eksempel 4.9

Uregelmæssig interkalation, måneder-dage indenfor 1 år (som strengt taget ikke kan kaldes en cyklus, da længden af året også varierer).

Tabel 4.9														
Uregelmæssig interkalation - 1 år														
(p, d) til t														
almindeligt år = 365 døgn														
	tilrettet cyklus 1							tilrettet cyklus 2						
	periode							periode						
p	jan	feb	mar	apr	maj	jun	jul	aug	sep	okt	nov	dec		
d	1	2	3	4	5	6	7	8	9	10	11	12		
1	1	32	60	91	121	152	182	213	244	274	305	335	resten	
2	2	33	61	92	122	153	183	214	245	275	306	336	bruges	
.....	ikke	
28	28	59	87	118	148	179	209	240	271	301	332	362		
29	29		88	119	149	180	210	241	272	302	333	363		
30	30		89	120	150	181	211	242	273	303	334	364		
31	31		90		151		212	243		304		365		
(p, d) til t														
skudår = 366 døgn														
	tilrettet cyklus 1							tilrettet cyklus 2						
	periode							periode						
p	jan	feb	mar	apr	maj	jun	jul	aug	sep	okt	nov	dec		
d	1	2	3	4	5	6	7	8	9	10	11	12		
1	1	32	61	92	122	153	183	214	245	275	306	336	resten	
2	2	33	62	93	123	154	184	215	246	276	307	337	bruges	
.....	ikke	
28	28	59	88	119	149	180	210	241	272	302	333	363		
29	29	60	89	120	150	181	211	242	273	303	334	364		
30	30		90	121	151	182	212	243	274	304	335	365		
31	31		91		152		213	244		305		366		

Så kan de 2 algoritmer stilles op:

Algoritme 3: DAGIÅR, Danmark og Norge

 /* D A G I Å R , 1. variant – med brug af årtype */

integer procedure dagiaar1(år,md,dg);
integer år,md,dg;
 /* beregner dagens nummer i året ud fra datoen,
 gælder for dansk kalender 600 – 3199 */

begin

integer korr;

/* beregn først korrektionen til dagnummeret */

korr := 3 - årtype(år);

if år = 1700

then korr := 12;

dagiaar1 := (214 * md - 211) // 7 + dg;

if dagiaar1 + korr > 61

then dagiaar1 := dagiaar1 - korr;

return dagiaar1;

end DAGIÅR1

Algoritme 4: DATOIÅR, Danmark og Norge

 /* D A T O I Å R , 1. variant – med brug af årtype */

procedure datoiaar1(år,dagiaar1,md,dg);

integer år,dagiaar1,md,dg;

/* beregner datoen i året ud fra dagens nummer,
 gælder for dansk kalender 600 – 3199 */

begin

integer korr;

/* beregn først korrektionen til dagnummeret */

korr := 3 - årtype(år);

if år = 1700

then korr := 12;

if dagiaar1 + korr > 61

then dagiaar1 := dagiaar1 + korr;

md := (dagiaar1 * 7 + 210) // 214;

dg := ((dagiaar1 * 7 + 210) mod 214) // 7 + 1;

return md,dg;

end DATOIÅR1

4. Interkalation – definition og eksempler

5. Den julianske dagtælling

At regne i antal dage

Meget ofte vil man være nødt til at regne i antal dage. Har man så kun enorme tabeller over skudår og månedernes forskellige længder i tankerne, kan man let synes at en brugbar algoritme bliver vanskelig at lave. Men også her kommer interkalationerne til hjælp.

For at løse den slags problemer er det lettest at operere med en **fortløbende dagtælling**, hvor hver dag har sit helt eget løbenummer totalt uafhængig af alt andet - det være sig kalender, årstal eller måned. Det eneste der skal fastlægges for en sådan tælling er datoen for løbenummer = 0. Man kan i princippet vælge frit, da det jo ikke er selve løbenumrene, men **differenserne** imellem dem som har interesse.

Imidlertid er der allerede en international standard for en fortløbende dagtælling, nemlig den julianske dagtælling*, og så kan man lige så godt holde sig til den. Enhver anden nummerering vil jo blot afvige med et konstant dagantal.

Den julianske dagtælling og dens historie

Den julianske dagtælling er et lille stykke kulturhistorie i sig selv; ideen om tællingen stammer tilbage fra renæssancen. Filologen og historikeren Joseph Justus Scaliger (1540-1609) foreslog den i sit værk: "De emendatione temporum" (Tanker om en forbedret tidsregning) fra 1583 [5]. Værket bliver i dag betragtet som banebrydende for den moderne, kritiske kronologi (læren om tidsregning).

Han opkaldte tællingen efter sin far, Julius Caesar Scaliger, og derved har navnet givet anledning til en del forvirring. Tællingen har med den julianske kalender at gøre, men er altså ikke opkaldt efter den.

Scaliger opstillede en cyklus på $28 \cdot 19 \cdot 15$ år = 7980 år à $365\frac{1}{4}$ dag i den julianske kalender*, dvs. i alt 2914695 dage. Så forestillede han sig man kunne tælle forfra igen. Af praktiske grunde vil man nu nok til sin tid bare tælle videre.

De 28 og 19 år kan man genkende som Solcirklen* - ugedagenes repetition og Metoncyklen*, mens de 15 år kommer fra indiktionen*, som i Romerriget blev anvendt til skatteudskrivning. Til dateringer blev indiktionen benyttet parallelt med de 2 andre, men den har altså ingen astronomisk oprindelse.

Da 28, 19 og 15 er indbyrdes primiske, (dvs. de har ingen tal fælles, der går op i dem andet end tallet 1), er 7980 år den mindste periode, efter hvilken både solcirkel, gyldental og indiktion igen = 1. Den kaldes den **julianske periode***.

Sidste gang det skete, var år 4713 f.v.t. Her skal man passe på, for det svarer til år -4712, jfr. bemærkningerne i kapitel 1.

Scaliger satte derfor dag nummer 0 til datoen -4712 jan 1, juliansk kalender. Fordelen ved det nulpunkt er at stort set alle kendte begivenheder har et positivt dagnummer. Perioden løber ud år +3268 jan 23, gregoriansk kalender.

Astronomerne tog den ret hurtigt i brug til langperiodiske fænomener, typisk variable stjerner med perioder på 200 dage og derover. Diagrammer over lysstyrke indeholder så blot **JD** som tidsangivelse, hvad der nok kan virke som en gåde for mange. Senere kom også historikerne til. De benytter tællingen som fælles reference for alle tidsregningerne i de forskellige kulturer.

5. Den julianske dagtælling

Der er en lille pudsighed ved tællingen: Det nye dagnummer starter kl 1200 UT. Det var et hensyn til astronomerne i Europa: Observationer gennem en europæisk nat skal have samme nummer! Man underinddeler tit tællingen, så man også får tidspunktet med i decimaler af et døgn, fx dagnummer = 2448882,430134 (= 1992 sep 16, kl. 22:19:23,5776, gregoriansk kalender). Når man regner i hele tal, plejer man traditionelt at angive dagnummer for en dato som det nummer, der begynder kl 12:00 UT.

Det julianske dagtal må ikke forveksles med dagens nummer indenfor et år, selvom den samme betegnelse desværre anvendes for begge begreber.

Analyse af omsætning mellem dagtælling og datoer

Den hårde knast er at få konstrueret algoritmer til denne omsætning. På grund af den julianske og gregorianske kalenders komplicerede cyklusser er der brug for flere interkalationer: år-dage, måneder-dage, århunder-dage. Der gøres udstrakt brug af mulighederne for at lægge flere interkalationer ovenover hinanden.

I kapitel 4 konstrueredes omsætteren mellem dato og dagnummer indenfor et år. På grund af februar var det dengang nødvendigt at indføre mindre modifikationer til den oprindelige omsætter. Det var til at leve med, når beregningerne holdt sig inden for et år.

Med flere år stiller sagen sig anderledes; her vil det være rart at undgå modifikationer - og det er da også heldigvis muligt.

Et trick: Året starter 1.marts!

Ved at lade året begynde 1.marts, får man januar og februar som de sidste måneder, nr 13 og 14. Når man så holder årenes længder på 365 og 366, bliver det muligt at konstruere en jævn interkalation, hvad der forenkler algoritmerne betydeligt. Feb 28 og feb 29 kommer så til sidst i vores beregningsår! Ideen stammer fra [1].

Da årstallet direkte er inddata, anvendes datoen: år 0 feb 29 som nulpunkt for begge kalendere. Så skal det julianske dagtal for denne dato blot regnes ud én gang for alle og lægges til eller trækkes fra i algoritmerne.

Eksempel 5.1 Juliansk kalender

Jævn interkalation år-dage i en 4-års cyklus.

Her kan man direkte benytte eksempel 4.7. Da en cyklus nu begynder med 1. marts i et skudår, skal ”skud-året” på 366 dage dog ligge sidst.

År delelige med 4 får nummer = 3 indenfor 4-års perioden, (fx år 0 mar 01 - år 3 feb 29).

Cyklus = 4 år = 1461 døgn, nu repeteret 365, 365, 365, 366; 365, ...

Periode = 1 år = 365 eller 366 døgn.

Enhed = 1 døgn.

A = 1461, B = 0, Q = 4, interkalationsformlerne bliver da:

$$u = (1461p) // 4 + d$$

$$p = (4u - 1) // 1461$$

$$d = ((4u - 1) \bmod 1461) // 4 + 1$$

5. Den julianske dagtælling

Tabel 5.1														
Jævn interkalation - år i den Julianske kalender														
(p, d) til u														
$u = (1461p) // 4 + d$														
p d	cyklus 0				cyklus 1				cyklus 2				cyklus 3 ►	
	periode				periode				periode				periode	
	0	1	2	3	4	5	6	7	8	9	10	11	12	osv
1	1	366	731	1096	1462	1827	2192	2557	2923	3288	3653	4018	4384	
2	2	367	732	1097	1463	1828	2193	2558	2924	3289	3654	4019	4385	
3	3	368	733	1098	1464	1829	2194	2559	2925	3290	3655	4020	4386	
.....
364	364	729	1094	1459	1825	2190	2555	2920	3286	3651	4016	4381	4747	
365	365	730	1095	1460	1826	2191	2556	2921	3287	3652	4017	4382	4748	
366				1461				2922				4383		

Eksempel 5.2 Juliansk kalender

Jævn interkalation måneder-dage i en års cyklus

Her er det muligt at lægge den 4-årige cyklus fra eksempel 5.1 ovenover denne cyklus. Derved behøver den nu givne cyklus **ikke at have en fast længde**, da 4-års cyklussen automatisk regulerer det. På den måde minder overvejelserne meget om dem i eksempel 4.8.

Den givne cyklus fra starten af = 1 år = 365 eller 366 døgn – det tager 4-års cyklussen sig af.

Periode = 1 måned på 31 eller 30 døgn, og så februar til sidst, som reguleres af 4-års cyklussen.

Repetitionsmønster med marts i starten: 31, 30, 31, 30, 31, 31, 30, 31, 30, 31, 31, feb. Det kan reduceres til en **mindre** cyklus på 5 måneder, så repetitionsmønstret bliver 31, 30, 31, 30, 31; 31, ...

Reduceret cyklus = 5 måneder = 153 døgn

Enhed = 1 døgn

A=153, B = -457, Q = 5. Den store negative værdi af B skyldes, at der skal begyndes med p (måned) = 3.

Interkalationsformlerne bliver:

$$u = (153p - 457) // 5 + d$$

$$p = (5u + 456) // 153$$

$$d = ((5u + 456) \bmod 153) // 5 + 1$$

Tabel 5.2														
Jævn interkalation - 1 år, med start i marts														
(p, d) til u														
$u = (153p - 457) // 5 + d$														
p d	cyklus 1					cyklus 2					cyklus 3 ►			
	periode					periode					periode			
	mar	apr	maj	jun	jul	aug	sep	okt	nov	dec	jan	feb	resten	
	3	4	5	6	7	8	9	10	11	12	13	14	bruges ikke	
1	1	32	62	93	123	154	185	215	246	276	307	338		
2	2	33	63	94	124	155	186	216	247	277	308	339		
.....		
28	28	59	89	120	150	181	212	242	273	303	334	365		
29	29	60	90	121	151	182	213	243	274	304	335	(366)		
30	30	61	91	122	152	183	214	244	275	305	336	ubrugt		
31	31		92		153	184		245		306	337			

5. Den julianske dagtælling

Eksempel 5.3 Gregoriansk kalender

Jævn interkalation 100 år - dage i en 400-års cyklus

På samme måde som i eksempel 5.1 er der brug for en interkalation, der arbejder med en 400-årig periode. Århundredet, som bevarer skuddagen, skal være sidst i perioden. Århundreder delelige med 4 får nummer = 3 indenfor 400-års perioden, (fx år 0 mar 01 - år 399 feb 29).

Cyklus = 400 år = 4 perioder = 146097 døgn

Periode = 36524 eller 36525 døgn.

Repetitions mønster = 36524, 36524, 36524, 36525; 36524, ...

Enhed = 1 døgn.

A = 146097, B = 0, Q = 4, interkalationsformlerne bliver:

$$u = (146097p) // 4 + d$$

$$p = (4u - 1) // 146097$$

$$d = ((4u - 1) \text{ mod } 146097) // 4 + 1$$

Tabel 5.3									
Jævn interkalation - 400-årig cyklus, Gregorianske kalender									
(p, d) til u									
u = (146097p)//4 + d									
		cyklus 0				cyklus 1			
		periode				periode			
d \ p		0	1	2	3	4	5	6	7
1		1	36525	73049	109573	146098	182622	219146	255670
2		2	36526	73050	109574	146099	182623	219147	255671
3		3	36527	73051	109575	146100	182624	219148	255672
.....	
36523		36523	73047	109571	146095	182620	219144	255668	292192
36524		36524	73048	109572	146096	182621	219145	255669	292193
36525					146097				292194

Eksempel 5.4 Gregoriansk kalender

Jævn interkalation år - dage i en 4-års cyklus

Interkalationen er den samme som i eksempel 5.1, bortset fra at år nummer 3 ikke altid er et skudår.

Det klarer interkalationen ovenover.

Eksempel 5.5 Gregoriansk kalender

Jævn interkalation - måneder-dage i en års-cyklus

Interkalationen er den samme som i eksempel 5.2. Heller ikke her spiller februars længde nogen rolle.

Forenkling af algoritmerne ved at slå omsætterne sammen

Algoritmerne kan forenkles betydeligt ved at slå interkalationerne for de 2 kalendere sammen. At der bliver indført en ekstra, lidt overflødig 100år-dage interkalation for den julianske kalender, gør ikke den store forskel. Antal dage på de 400 år skal da blot = 146100.

Årstallene er de astronomiske. Hvis man har brug for de historiske årstal, skal man for astronomisk år < 1 benytte omskrivningen historisk år f.v.t = (1 - astronomisk år).

Algoritme 5: JDAG, Danmark og Norge

```

-----
/* J D A G */

integer procedure jdag(år,md,dg);
integer år,md,dg;
/* Beregner juliansk dagtal for datoen,
   gælder for dansk-norsk kalender 600 – 3199,
   jdag = 0 svarer til -4712 jan 01, juliansk eller til -4713 nov 24, gregoriansk */

begin
integer totdato, nulldag, cyklgd, cykantal, åricyk, århunicyk;

/* beregn nulldag og cykisk længde for begge kalendere,
   skæringsdatoer for Danmark-Norge: 1700 feb 18 - 1700 mar 01 */

totdato := år * 100 + md;
if totdato > 170002
then begin                                     /* gregoriansk kalender */
    nulldag := 1721119; cyklgd := 146097
end
else begin                                     /* juliansk kalender */
    nulldag := 1721117; cyklgd := 146100
end;

/* januar og februar regnes som hørende til året før */
if md < 3
then begin
    md := md + 12; år := år - 1
end;

cykantal := entier(år / 400);                    /* antal cyklusser à 400 år */

åricyk := år - cykantal * 400;                  /* årets placering i 400 år */
århunicyk := åricyk // 100;                    /* tilhørende hundredår */
årrest := åricyk mod 100;                     /* årets placering i 100 år */

jdag := nulldag + cykantal * cyklgd             /* find placering fra år 0 */
+ (århunicyk * cyklgd) // 4 + (årrest * 1461) // 4 /* og addér de 3 jævne */
+ (153 * md - 457) // 5 + dg;                  /* interkalationer */

return jdag;

end JDAG
-----

```

Algoritme 6: DATOJ, Danmark og Norge

```

-----
/* D A T O J */

procedure datoj(jdagnr,år,md,dg);
integer jdagnr;
/* beregner dato for et juliansk dagtal,
   gælder for dansk norsk kalender 600 – 3199,
   jdag = 0 svarer til -4712 jan 01, juliansk, eller til -4713 nov 24, gregoriansk */

begin
integer nulldag,cyklgd,cykantal,dagicyk,dagiårhun,århunicyk,årrest,dagrest;

/* beregn nulldag og cyklgd for den relevante kalender,
   skæringsdatoer for Danmark: 1700 feb 18 - 1700 mar 01 */

if jdagnr > 2342031
then begin                                     /* gregoriansk kalender */
  nulldag := 1721119; cyklgd := 146097
end
else begin                                     /* juliansk kalender */
  nulldag := 1721117; cyklgd := 146100
end;

jdagnr := jdagnr - nulldag;                       /* relativt til år 0 feb 29 */

cykantal := entier((jdagnr - 1) / cyklgd);        /* antal cyklusser à 400 år */
dagicyk := jdagnr - cykantal * cyklgd;

århunicyk := (4 * dagicyk - 1) // cyklgd;         /* århundrede i cyklus */
dagiårhun := ((4 * dagicyk - 1) mod cyklgd) // 4 + 1; /* dagantal i 100 år */

årrest := (4 * dagiårhun - 1) // 1461;           /* år i 4-års cyklus */
dagrest := ((4 * dagiårhun - 1) mod 1461) // 4 + 1; /* dagantal i 4 år */

md := (5 * dagrest + 456) // 153;                /* måned, dag og år */
dg := ((5 * dagrest + 456) mod 153) // 5 + 1;
år := cykantal * 400 + århunicyk * 100 + årrest;

/* ret årstal for januar og februar */
if md > 12
then begin
  md := md - 12; år := år + 1
end;

return år,md,dg;

end DATOJ
-----

```

Antal dage

Når problemerne med omregning af datoer til en fortløbende dagtælling er løst, kan målet let nås i dette kapitel, nemlig at kunne regne i antal dage. Det er blot at omregne datoer til den julianske dagtælling, regne i den, og evt. omregne igen.

Algoritme 7: DATODIFF, Danmark og Norge

```
/* D A T O D I F F */  
  
integer procedure datodiff(år1,md1,dg1,år2,md2,dg2);  
integer år1,md1,dg1,år2,md2,dg2;  
  
/* beregner antal dage mellem 2 datoer med fortegn,  
   gælder for kalenderen i Danmark-Norge 600 – 3199,  
   hvis (år1,md1,dg1) er før (år2,md2,dg2), er datodiff > 0,  
   tidligste dato regnes med, seneste ikke */  
  
begin  
  datodiff := jdag(år2,md2,dg2) - jdag(år1,md1,dg1);  
  
return datodiff;  
  
end DATODIFF
```

Algoritme 8: NYDATO, Danmark og Norge

```
/* N Y D A T O */  
  
procedure nydato(år,md,dg,dagantal,nyår,nymd,nydg);  
integer år,md,dg,dagantal,nyår,nymd,nydg;  
  
/* beregner en ny dato ud fra en given dato og et antal dage,  
   gælder for kalenderen i Danmark-Norge 600-3199,  
   skæringsdatoer for Danmark: 1700 feb 18 - 1700 mar 01 */  
  
begin  
  integer jtal;  
  
  jtal := jdag(år,md,dg);  
  datoj(jtal + dagantal,nyår,nymd,nydg);  
  
return nyår, nymd, nydg;  
  
end NYDATO
```

Ny omsætning mellem dato og dagnummer i et år

Med de julianske dagtal er der en anden måde at omsætte mellem dato og dagtal i et år.

Man får 2 nye varianter af algoritmerne næsten gratis ved at kigge på forskellen mellem en dato og jan 01 det år.

Algoritme 3.2, DAGIÅR, Danmark og Norge

```
/* D A G I Å R , variant 2 – med brug af jdag */
```

```
integer procedure dagiår2(år,md,dg);
```

```
integer år,md,dg;
```

```
/* beregner dagens nummer i året udfra datoen,  
gælder for Danmark-Norge 600 – 3199 */
```

```
begin
```

```
dagiår2 := jdag(år,md,dg) - jdag(år,1,1) + 1;
```

```
return dagiår2;
```

```
end DAGIÅR2
```

Og den omvendte:

Algoritme 4.2, DATOIÅR, Danmark og Norge

```
/* D A T O I Å R , variant 2 – med brug af dato_j */
```

```
procedure datoiår2(år,dagiår,md,dg);
```

```
integer år,dagiår,md,dg;
```

```
/* beregner datoen i året udfra dagens nummer,  
gælder for Danmark-Norge 600 – 3199 */
```

```
begin
```

```
integer resjdagnr;
```

```
resjdagnr := jdag(år,1,1) + dagiår - 1;
```

```
datoj(resjdagnr,år,md,dg);
```

```
return md,dg;
```

```
end DATOIÅR2
```

6. Ugenummer og ugedag

Inden for de senere år er der kommet et nyt element til ved beregninger af datoer, nemlig ugenumrene. Mange aftaler om produktion og leverancer er nu baseret på dateringer af typen "onsdag i uge 42". Der er derfor også brug for algoritmer, som kan regne datoer om til ugenummer og ugedag og omvendt.

Definition af ugenumrene

DS/ISO standarden nr 8601 [19] definerer ugenumrene i afsnit 4.3.2.2 på denne måde:

1. En uge omfatter altid et tidsrum på 7 dage.
2. Mandag er den første dag i ugen.
3. Uge nr 1 i et år er den første uge, som indeholder den første torsdag i det nye år.

Konsekvenser af definitionen

Man kan af definitionen umiddelbart slutte:

- Der er ikke noget, der hedder uge 0 i et år.
- 4. januar ligger altid i uge 1.
 - hvis den er mandag, begynder uge 1 med 4.januar
 - hvis den er søndag, slutter uge 1 med 4.januar
- Datoer i et år som ligger før uge 1 - det kan højst dreje sig om 3 dage - har et ugenummer som er fastsat af året før.
- Datoer sidst i året - igen højst 3 dage - kan ligge i uge 1 året efter.

Oversigt over mulighederne

For at få lidt mere hold på ugenumrene ved årsskifterne, er det rart med en oversigt som viser de mulige kombinationer. Det er i øvrigt en metode, der generelt kan anbefales ved kalenderberegninger – eller som én af forfatterens kolleger meget rammende har udtrykt det: ”**Opskriv hele udfaldsrummet**”.

Der er 7 muligheder for det pågældende år, og hver kan kombineres med 2 muligheder for året før - i alt 14 tilfælde. Se skemaerne nedenfor, hvor ugenumre er sat foran datoerne.

6. Ugenummer og ugedag

Ugedagskombinationer ved årsskifter

Tabel 6.1												
Ugenumre, ugedagskombinationer ved årsskifter												
	Alm.år		Skudår		Alm.år		Skudår		Alm.år		Skudår	
	uge	dato	uge	dato	uge	dato	uge	dato	uge	dato	uge	dato
året før,	1	M 1	*	S 1	1	Ti 1	1	M 1	1	O 1	1	Ti 1
		Ti 2	1	M 2		O 2		Ti 2		To 2		O 2
januar		O 3		Ti 3		To 3		O 3		F 3		To 3
		To 4		O 4		F 4		To 4		L 4		F 4
	1	F 5	1	To 5	1	L 5	1	F 5	1	S 5	1	L 5
-----	-----		-----		-----		-----		-----		-----	
december	52	O 26			52	To 26			52	F 26		
		To 27				F 27				L 27		
		F 28				L 28			52	S 28		
		L 29			52	S 29			1	M 29		
	52	S 30			1	M 30				Ti 30		
	1	M 31			1	Ti 31			1	O 31		
året nu,	1	Ti 1			1	O 1			1	To 1		
		O 2				To 2				F 26		
januar		To 3				F 3				L 3		
		F 4				L 4			1	S 4		
	1	L 5			1	S 5			2	M 5		
	Alm.år		Skudår		Alm.år		Skudår		Alm.år		Skudår	
	uge	dato	uge	dato	uge	dato	uge	dato	uge	dato	uge	dato
året før,	*	F 1	1	To 1	*	L 1	*	F 1	*	S 1	*	L 1
		L 2		F 2		S 2		L 2	1	M 2		S 2
januar	*	S 3		L 3	1	M 3	*	S 3		Ti 3	1	M 3
	1	M 4	1	S 4		Ti 4	1	M 4		O 4		Ti 4
	1	Ti 5	2	M 5	1	O 5	1	Ti 5	1	To 5	1	O 5
-----	-----		-----		-----		-----		-----		-----	
december	51	S 26	52	S 26	52	M 26			52	Ti 26		
	52	M 27	53	M 27		Ti 27				O 27		
		Ti 28		Ti 28		O 28				To 28		
		O 29		O 29		To 29				F 29		
		To 30		To 30		F 30				L 30		
	52	F 31	53	F 31	52	L 31			52	S 31		
året nu,	52	L 1	53	L 1	52	S 1			1	M 1		
	52	S 2	53	S 2	1	M 2				Ti 2		
januar	1	M 3	1	M 3		Ti 3				O 3		
		Ti 4		Ti 4		O 4				To 4		
	1	O 5	1	O 5	1	To 5			1	F 5		

6. Ugenummer og ugedag

Af tabellen kan man se i et år, at evt. dage før uge 1 ligger i:

- uge 53, når **året før** begyndte med:
 - torsdag 1. januar (fx. i årene: 1977, 1982, 1988, 1999, 2005, 2010, 2016, 2027)
 - onsdag 1. januar og var skudår (fx. i årene: 1993 og 2021)
- uge 52 ellers

Som kuriosum skal det lige bemærkes, at overgangsåret i Danmark, år 1700, kun havde 355 dage og ugenumrene skulle så gå fra 1 til 51. Da det i den gregorianske (ny) kalender var lørdag 1. januar 1701, skal ugenummeret for 1. og 2. januar 1701 altså være 51. Men definitionen er næppe relevant så langt tilbage.

Læg iøvrigt mærke til, når et år begynder med lørdag 1. januar, så er det uge 53 efter et skudår, ellers uge 52.

Jævne interkalationer kan godt benyttes til alene at bestemme, om største ugenr for et år er 52 eller 53, men da man skal have fat i ugedagene, må den julianske dagtælling med i algoritmerne.

Ugebasisdato for et år

Ugenummereringen er derfor noget kompliceret i forhold til årsskifterne. En brugbar løsning til problemet er at fastlægge en **ugebasisdato** for hvert år, så ugenummer og ugedag kan beregnes ved at tage antallet af dage mellem den angivne dato og ugebasisdatoen.

Ugebasisdato for et år må **ikke** forveksles med basis(dato) for et år, som er dagen før 1. marts i året, jfr kapitel 8.

Da mandag er udnævnt til at være den første dag i en ny uge, og da det er hensigtsmæssigt at tælle fra nul, defineres ugebasisdatoen for et år sådan:

ugebasisdato for et år = mandag i ugen **før** uge 1 for året (50)

Ugebasisdatoen er udgangspunktet – **nuldagen i nulugen** - for ugenummereringen i et år, og det giver en naturlig beregning af ugenummer og ugedag, da de bliver henholdsvis heltalskvotient og rest ved division med 7 af dagantallet fra ugebasisdatoen til den forelagte dato.

Og med den definition bliver:

Største ugenummer for et år

= (dagantallet mellem ugebasisdato for året efter og ugebasisdato for selve året) // 7 (51)

da der altid er et helt antal uger mellem 2 ugebasisdatoer.

De eneste problemer opstår, hvis den forelagte dato ligger inden for én af de 2 kritiske intervaller ved nytårsskifterne, 3 dage på hver side af nytårene. Hvis det foreløbig beregnede ugenummer så er mindre end 1, skal det korrigeres til største ugenummer for året før, og hvis det er større end største ugenummer for året, skal det korrigeres til ugenummer 1 for året efter.

Når beregningen af ugenummer skal være generel, er man **altid** nødt til at **beregne 2 ugebasisdatoer** - for året nu og enten for året før eller året efter. Ellers vil der fx optræde fejl ved et afkortet overgangsår, (Danmark år 1700, Sverige år 1753).

Der er en lille asymmetri mellem de 2 kritiske intervaller. Ved det første nytårsskifte er det nok at undersøge om ugenummer < 1, så skal ugebasisdato for året før også beregnes, mens man ved det andet er nødt til først at undersøge datointervallet.

6. Ugenummer og ugedag

Problemet er så at få beregnet ugebasisdatoen for et år. Her kommer en oversigt over de 7 muligheder:

Tabel 6.2 Beregning af ugebasisdato for et år							
Ugedag for jan 1 i året nu	mandag	tirsdag	onsdag	torsdag	fredag	lørdag	søndag
Mandag i uge 1, i året før eller i året nu	jan 1	dec 31	dec 30	dec 29	jan 4	jan 3	jan 2
Ugebasisdato for året, altid i året før	dec 25	dec 24	dec 23	dec 22	dec 28	dec 27	dec 26
Dagantal mellem ugebasisdato og jan 1	7	8	9	10	4	5	6

Så føjes der 4 rækker mere til skemaet – alle data gælder for jan 1 i året nu:

Tabel 6.3 Beregning af ugebasisdato for et år - fortsat							
Ugedag for jan 1 i året nu	mandag	tirsdag	onsdag	torsdag	fredag	lørdag	søndag
Rå ugedagskode = jdagtal mod 7	0	1	2	3	4	5	6
Rå ugedagskode + 3	3	4	5	6	7	8	9
(Rå ugedagskode + 3) mod 7	3	4	5	6	0	1	2
(Rå ugedagskode + 3) mod 7 + 4	7	8	9	10	4	5	6

Dvs.

Antal dage mellem ugebasisdato og jan 1 for et år
= (rå ugedagskode + 3) mod 7 + 4

(52)

Den rå ugedagskode tæller fra 0 til 6 (mandag til søndag), mens den almindelig benyttede ugedagskode tæller fra 1 til 7, se fx i køreplaner.

Nu kan algoritmen stilles op:

Algoritme 9, UGE, Danmark og Norge

```

-----
/* U G E */

procedure uge(år,md,dg,ugenr,ugedagkod);
integer år,md,dg;

/* beregner ugenummer og kode for ugedag for en dato. Ugedagkod: 1=mandag, 2=tirsdag, ..., 7=søndag.
   Gælder for Danmark-Norge 600-3199, juliansk kalender før 1700, gregoriansk kalender efter.
   Ugenummer efter DS ISO-standarden:
   1. Et ugenummer omfatter altid et tidsrum på 7 dage
   2. Mandag er første dag i ugen
   3. Uge nr 1 i et år er den første uge, der indeholder mindst 4 dage af det nye år */

begin
integer jtal,jtalfør,jtalnu,jtalefter,jugebasisfør,jugebasisnu,jugebasisefter,maxugefør, maxugenu;

/* beregn juliansk dagtal for den aktuelle dato, jan 01 for året nu, og ugebasisdato for året */
jtal := jdag(år,md,dg);
jtalnu := jdag(år,1,1);
jugebasisnu := jtalnu - ((jtalnu + 3) mod 7 + 4);
/* beregn foreløbigt ugenummer, og ugedag */
ugenr := (jtal - jugebasisnu) // 7;
ugedagkod := (jtal - jugebasisnu) mod 7 + 1;

if ugenr < 1
then begin
  /* datoen ligger i det 1. kritiske interval omkring nytår for året, ugenr skal rettes,
     beregn for året før: juliansk dagtal for jan 01, ugebasisdato og størst mulige ugenummer */
  jtalfør := jdag(år - 1,1,1);
  jugebasisfør := jtalfør - ((jtalfør + 3) mod 7 + 4);
  maxugefør := (jugebasisnu - jugebasisfør) // 7;
  ugenr := maxugefør;
end
else begin
  /* datoen kan ligge i det 2. kritiske interval omkring nytår for året efter,
     beregn for året efter: juliansk dagtal for jan 01 */
  jtalefter := jdag(år + 1,1,1);
  /* undersøg det 2. kritiske interval */
  if (jtal - jtalefter) > -4 and (jtal - jtalefter) < 3
  then begin
    /* datoen ligger i det 2. kritiske interval,
       beregn for året efter: ugebasisdato, og for året nu: størst mulige ugenummer */
    jugebasisefter := jtalefter - ((jtalefter + 3) mod 7 + 4);
    maxugenu := (jugebasisefter - jugebasisnu) // 7;
    /* ret evt. ugenr */
    if ugenr > maxugenu
    then ugenr := 1;
  end;
end;

return ugenr, ugedagkod;

end UGE
-----

```

6. Ugenummer og ugedag

Der er også brug for en omvendt algoritme, som kan beregne datoen for et givet ugenummer og ugedag i et år. Her skal man beregne ugebasisdato for året selv og året efter.

Algoritme 10, UGESTART, Danmark og Norge

/* U G E S T A R T */

procedure ugestart(årind, ugenr, ugedagkod, år, md, dg);
integer årind, ugenr, år, md, dg;

/* beregner datoen for et ugenummer og en ugedag i et år,
denne dato kan ligge i et andet år end det givne!
Ugenummer efter DS ISO-standarden:
1. Et ugenummer omfatter altid et tidsrum på 7 dage
2. Mandag er første dag i ugen
3. Uge nr 1 i et år er den første uge, der indeholder mindst 4 dage af det nye år */

begin

integer jtal, jtalnu, jtalefter, jugebasisnu, jugebasisefter, maxugefør, maxugenu;

/* indledende kontrol af ugenummer */

if ugenr < 1 **or** ugenr > 53

then quit; /* fejlreaktion */

/* beregn juliansk dagtal for jan 01 for året selv og året efter */

jtalnu := jdag(år,1,1);

jtalefter := jdag(år + 1,1,1);

/* beregn juliansk dagtal for ugebasisdatoen for året selv og året efter */

jugebasisnu := jtalnu - ((jtalnu + 3) **mod** 7 + 4);

jugebasisefter := jtalefter - ((jtalefter + 3) **mod** 7 + 4);

/* beregn størst mulige ugenummer for året nu */

maxugenu := (jugebasisefter - jugebasisnu) // 7;

/* afsluttende kontrol af ugenummer, ingen fejlagtig uge 53 */

if ugenr > maxugenu

then quit; /* fejlreaktion */

/* beregn juliansk dagtal for den ønskede dato, og omregn den til datoen */

jtal := jugebasisnu + ugenr * 7 + ugedagkod - 1;

datoj(jtal, år, md, dg);

return år,md,dg;

end UGESTART

7. Beregningen af Påskedag

I den kristne kalender er Påskedag nøglen til de forskydelige helligdage om foråret. I Danmark gælder det for: Fastelavn, Påsken, Store Bededag, Kristi Himmelfartsdag og Pinsen.

Kirkeåret

I kirkelig henseende styrer placeringen af Påskedag langt den største del af kirkeåret, nemlig tiden fra søndag Septuagesima, 2 uger før Fastelavn, til kirkeårets slutning ved sidste søndag efter Trinitatis i november. Trinitatis selv ligger 1 uge efter Pinsedag, og fejres kun i de protestantiske kirker.

Begyndelsen af kirkeåret, der indledes med 1. søndag i Advent i november eller december og omfatter Jul, Nytår og Hellig 3 Konger, styres af ugedagene i de pågældende år.

Placeringen af Påskedag

Som nævnt i kapitel 1 blev Påskedagen ved kirkemødet i Nikæa år 325 fastlagt til **første søndag efter første fuldmåne efter forårsjævndøgn**, og mødet overlod til patriarken/biskoppen af Alexandria at beregne og meddele det til hele kristenheden for at få gjort en ende på den forvirring der var opstået i oldkirken. Bl.a. fejrede de jødekristerne oprindeligt Påsken sammen med jøderne, mens de hedningekristne var enige om at ugedagene også skulle respekteres: Kristus blev korsfæstet på en fredag og søndagen derefter var Påskedag med opstandelsen [5].

I den forbindelse skal man først og fremmest huske at selv denne fastlæggelse er cyklisk*, dvs. gennemsnitlig, så kirken kunne forudsige Påskens placering i fremtiden. Det var mest hensigtsmæssigt med datidens langsomme kommunikation. Ellers kunne man komme ud for mangt og meget, bl.a. at det astronomiske fuldmåneøjeblik samtidig faldt på en dato lige efter midnat i øst og dagen før sent om aftenen i vest med derved følgende mulighed for 2 forskellige datoer for Påskedag.

Men selv med denne fastlæggelse var der uenighed om, hvornår Påsken skulle fejres i den julianske kalender:

- Efter **hvilken cyklus** skulle månefaserne beregnes? Nogle tidlige kristne forsøgte sig med en 8-årig, der lynhurtigt viste sig alt for upræcis i forhold til Cæsars kalenderår, de romerske kirker prøvede med en 84-årig; den var lidt bedre, mens alexandrinerne benyttede Metoncyklen* på 19 år, som viste sig de andre overlegen i præcision [5] [6].
- Hvis nu **første fuldmåne** efter forårsjævndøgn faldt på en **lørdag**, ville nogle udskyde tidspunktet for den korrekte fuldmåne af hensyn til at overholde fasten i fuld længde, mens andre var tilfredse med ”14.dagen efter nymånen”. Hvad var korrekt?
- Hvis **første fuldmåne** faldt på en **søndag**, skulle den søndag så være Påskedag eller skulle Påsken udskydes en uge?
- Hvilke **ydergrænser** skulle gælde for Påskedag? I vest regnede man oprindeligt med 25.marts – 21.april, romerne var ikke glade for senere datoer på grund af en gammel hedensk fest der faldt på 21.april og fordi forårsjævndøgnet ved år 0 lå 24./25.marts, mens alexandrinerne regnede ydergrænserne fra 22.marts – 25.april, og år 325 lå jævndøgnet 20./21. marts.

Alexandrinernes beregning slog efterhånden igennem – bl.a. kraftig støttet af autoriteter som den engelske munk Beda (ca 673-735). Det tog dog sin tid - først omkring 500 var beregningen alment anerkendt, og fra år 750 var den rettesnor i hele (den europæiske) kristenhed – både vest- og østkirken - efter disse regler:

- Kalenderen var den julianske med skudår hver 4.år og forårsjævndøgn fastlagt til 21.marts.
- Metoncyklen på 19 år skulle bruges.
- Fuldmåne på en lørdag kunne godt bruges, så Påskedag blev dagen efter.
- Fuldmåne på en søndag bevirkede, at Påskedag skulle udskydes til søndagen efter.
- Ydergrænserne for Påskedag blev 22.marts – 25.april.

7. Beregningen af Påskedag

Søndagsbogstaver og solcirkel

Til brug for ugedagsberegningen indførte man **søndagsbogstaver***, hvor man angav søndag 1.januar med A, søndag 2. januar med B, osv op til søndag 7. januar med G.

Oprindeligt blev der tildelt et bogstav til hver dato i året, og en evt. skuddag mellem 24. og 25. februar indgik også i tildelingen. Derved forrykkedes søndagsbogstaverne ligesom ugedagene for resten af året i skudår.

I mere nutidig udgave nøjes man med at tildele et bogstav til et helt år, nemlig bogstavet for 1.søndag i året. Hvis året er et skudår, skal det have to søndagsbogstaver, ét for januar-februar, og ét for marts-december. Det sidste skal gælde som om året ikke var skudår, fx vil et skudår med søndag 7.januar have søndagsbogstaverne GF, hvor marts-december så går efter et almindeligt år med bogstav F.

Som betegnelse dukker søndagsbogstav først op omkring år 800, men man havde opereret med begrebet i hvert fald lige siden Nikæamødet.

I den julianske kalender gentager ugedagene sig efter $7 \cdot 4$ år = 28 år. Denne periode kaldes **Solcirklen*** (dies solis = søndag) og i den julianske kalender er der et fast forhold mellem søndagsbogstav og solcirkel, mens den gregorianske kalender på grund af de nye skudårsregler ikke har den.

årets type ugedag for 1.januar	årets type	
	alm.år	skudår
søndag	A	AG
mandag	G	GF
tirsdag	F	FE
onsdag	E	ED
torsdag	D	DC
fredag	C	CB
lørdag	B	BA

Juliansk Påskedag: 19-årig månecyklus og 28-årig ugedagscyklus

Med Metoncyklen og Søndagsbogstaverne kunne man så tabellægge Påskedag. Hvert år i den 19-årige Metoncyklus fik et nummer mellem 1 og 19 i cyklussen, årets **gyldental***. Selve ordet dukker dog lige som søndagsbogstav først op flere århundreder senere.

Betegnende for datidens manglende regnefærdighed varede det længe, før man opdagede at Påskedagen i den julianske kalender vil gentage sig efter $28 \cdot 19$ år = 532 år. Det ser nærmest ud til at man 'snublede' over perioden omkring år 500 [6].

Juliansk epakt

Til hjælp for beregningen af Påskedag blev der i middelalderen ved siden af gyldental og søndagsbogstav efterhånden udviklet en række hjælpemidler, hvor **epakten*** nok er den vigtigste. Man kunne godt have klaret sig med gyldentallet, da det jo indirekte angiver de cykliske månefasen.

7. Beregningen af Påskedag

Epakt angiver månens alder - dvs. hvor mange dage der er gået efter sidste nymåne – for en dato.

I den julianske kalender kom man efterhånden frem til 2 forskellige epakter:

- Én der gælder for 22.marts, (Beda-epakten)
- Én der gælder for 1.januar, (Alexandrinsk epakt)

Epakten vokser med 11 hver år – men indenfor de 30 dage, som en synodisk måned varer. Man lagde et mønster af skiftevis 30 og 29 dage ned over alle datoer i et år for at tilnærme den synodiske måned* på ca 29½ døgn så godt som muligt til de astronomiske* månefasen.

Man tilpassede de 235 måne-måneder med 228 måne-måneder på skiftevis 29 og 30 dage og tilføjede (interkalerede) så 7 skudmåneder à 30 dage på passende steder. Dertil kommer $4\frac{3}{4}$ skuddage på de 19 julianske år.

Det bliver i alt til $6940\frac{3}{4}$ dage, mens 19 julianske år indeholder $6939\frac{3}{4}$ dage, og derfor skulle Månen ”springe” (saltus Lunae) 1 gang i løbet af de 19 år, dvs. epakten skal blive 1 større. Det spring lagde man til sidst i den 19-årige cyklus.

Nogle oversigter angiver epakt 30 som epakt 0 – traditionelt blev den i sin tid angivet med en stjerne * i tabellerne, hvor man anvendte romertallene.

Forholdet mellem gyldental og epakter ligger fast i den julianske kalender. Månens spring i epakterne kan ses fra gyldental 19 til gyldental 1.

Tabel 7.2 Juliansk gyldental og epakt																			
Gyldental = år mod 19 + 1																			
Beda epakt [mar 22] = (11*gyldental +18) mod 30 + 1																			
Alexandrinsk epakt [jan 01] = (11*gyldental - 4) mod 30 + 1																			
Gyldental	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Beda epakt	30	11	22	3	14	25	6	17	28	9	20	1	12	23	4	15	26	7	18
Alex. Epakt	8	19	30	11	22	3	14	25	6	17	28	9	20	1	12	23	4	15	26

Den julianske Påskedag kan da beregnes med søndagsbogstav, og enten gyldental eller én af epakterne. Tabellen er bearbejdet fra [5].

De til gyldentalene svarende epakter står på de tilsvarende pladser i tabellen – derfor den lidt sære rækkefølge af dem.

7. Beregningen af Påskedag

Tabel 7.3 Juliansk Påskedag												
Søndags- bogstav	Gyldental					Påskedag	Beda epakt					Alexandrinsk epakt
D, ED	16					mar 22	15					23
	2	5	10	13		mar 29	11	14	9	12		19 22 17 20
	4	7	12	15	18	apr 5	3	6	1	4	7	11 14 9 12 15
	1	6	9	17		apr 12	30	25	28	26		8 3 6 4
	3	8	11	14	19	apr 19	22	17	20	23	18	30 25 28 1 26
E, FE	5	16				mar 23	14	15				22 23
	2	10	13	18		mar 30	11	9	12	7		19 17 20 15
	1	4	7	12	15	apr 6	30	3	6	1	4	8 11 14 9 12
	6	9	14	17		apr 13	25	28	23	26		3 6 1 4
	3	8	11	19		apr 20	22	17	20	18		30 25 28 26
F, GF	5	16				mar 24	14	15				22 23
	2	7	10	13	18	mar 31	11	6	9	12	7	19 14 17 20 15
	1	4	12	15		apr 7	30	3	1	4		8 11 9 12
	3	6	9	14	17	apr 14	22	25	28	23	26	30 3 6 1 4
	8	11	19			apr 21	17	20	18			25 28 26
G, AG	5	13	16			mar 25	14	12	15			22 20 23
	2	7	10	18		apr 1	11	6	9	7		19 14 17 15
	1	4	9	12	15	apr 8	30	3	28	1	4	8 11 6 9 12
	3	6	14	17		apr 15	22	25	23	26		30 3 1 4
	8	11	19			apr 22	17	20	18			25 28 26
A, BA	2	5	13	16		mar 26	11	14	12	15		19 22 20 23
	7	10	15	18		apr 2	6	9	4	7		14 17 12 15
	1	4	9	12		apr 9	30	3	28	1		8 11 6 9
	3	6	11	14	17	apr 16	22	25	20	23	26	30 3 28 1 4
	8	19				apr 23	17	18				25 26
B, CB	2	5	13	16		mar 27	11	14	12	15		19 22 20 23
	4	7	10	15	18	apr 3	3	6	9	4	7	11 14 17 12 15
	1	9	12	17		apr 10	30	28	1	26		8 6 9 4
	3	6	11	14		apr 17	22	25	20	23		30 3 28 1
	8	19				apr 24	17	18				25 26
C, DC	2	5	10	13	16	mar 28	11	14	9	12	15	19 22 17 20 23
	4	7	15	18		apr 4	3	6	4	7		11 14 12 15
	1	6	9	12	17	apr 11	30	25	28	1	26	8 3 6 9 4
	3	11	14	19		apr 18	22	20	23	18		30 28 1 26
	8					apr 25	17					25

Gregoriansk Påskedag: Korrektioner med sol- og månejævning

Ved den katolske kirkes kalenderreform i 1582 rettede man så vidt muligt de 2 fejl der havde hobet sig op, men lagde samtidig meget stor vægt på så vidt mulig at holde fast i traditionen fra 325 og den julianske påskeberegning, naturligvis med det klare politiske formål at gøre reformen spiselig for de andre kirker – og derved indirekte fastslå romerkirkens faglige og teologiske overlegenhed (!) Man må huske på, at det var tiden for de store religionsstridigheder og -krige i Europa.

- Forårsjævndøgnet skal fortsat være - og holdes så tæt som muligt - på 21. marts.
- Påskedag skal fortsat være første søndag efter første fuldmåne derefter. Den cykliske fuldmåne skal fortsat holdes tæt på den astronomiske fuldmåne.
- Ydergrænserne for Påskedag er også stadig 22. marts – 25. april.

7. Beregningen af Påskedag

Til beregning af månefaserne benyttes nu den **gregorianske epakt***, der for et år angiver månens alder (antal dage fra sidste nymåne) på den 1. januar. Gyldentallet* kan naturligvis fortsat beregnes, men har tabt sin nøglerolle.

På samme måde som i den julianske kalender har man lagt et mønster af skiftevis fulde* og hule* måne-måneder på 30 og 29 dage nedover alle datoer i et år. Da der er 30 mulige epakter, anvender man en **dobbeltepakt 26-25**, normalt noteret som **(25)** i måne-måneder på 29 dage.

Sammen med det gregorianske søndagsbogstav fastlægger epakten den gregorianske Påskedag.

I den gregorianske kalender er der indført 2 korrektioner, på dansk kaldet **jævninger**:

- **Soljævning**, hvor der fjernes 3 skuddage på 400 år. Det er den mest kendte korrektion, der skal sikre at det astronomiske forårsjævndøgn og dermed årstiderne ikke igen forrykker sig fra 21. marts. Soljævningen medfører, at nymånen kommer 1 dag senere i kalenderen, dvs. Månens alder bliver mindre og derfor bliver epakterne også 1 mindre. Den foregår i 1700, 1800, 1900; 2100, 2200, 2300; 2500,
- **Månejævning**, hvor der over 2500 år korrigeres 8 gange, så de cykliske månefasen ikke forrykker sig fra de astronomiske. Månejævningen medfører, at nymånen kommer 1 dag tidligere, dvs. Månens alder bliver større og derfor bliver epakterne 1 større. Den foregår i 1800, 2100, 2400, 2700, 3000, 3300, 3600, 3900, 4300; 4600,

Sol- og månejævning går hver sin vej, og for at undgå skæve korrektioner skal de finde sted ved overgang fra et hundredeårstal til det næste (1899/1900, 1999/2000, 2099/2100 osv.).

Soljævning og månejævning, jævn interkalation

Med disse definitioner viser det sig, at begge korrektioner kan beregnes som jævne interkalationer!

Soljævning: **A = 3, B = -45, Q = 4**

Månejævning: **A = 8, B = -112, Q = 25**

Inden for et hundredår er der fortsat en fast relation mellem gyldental og epakt, men den vil udover de tidsrum være underkastet jævningerne. Konsekvensen af det bliver, at epakten forrykker sig i forhold til gyldentallet i de forskellige hundredår.

I 1500-tallet svarer gyldental 1 til gregoriansk epakt 1. I tabel 7.4 kan man se, hvordan det udvikler sig.

Tabel 7.4 Gregoriansk kalender, sol- og månejævning, udgangstabel for epakt				
Soljævning = $(3 \cdot (\text{år} // 100) - 45) // 4$				
Månejævning = $(8 \cdot (\text{år} // 100) - 112) // 25$				
Hundredårs- interval	sol- jævning	måne- jævning	samlet jævning	epakt, svarende til gyldental=1
1582 - 1599	0	0	0	1
1600 - 1699	0	0	0	1
1700 - 1799	-1	0	-1	0
1800 - 1899	-2	1	-1	0
1900 - 1999	-3	1	-2	29
2000 - 2099	-3	1	-2	29
2100 - 2199	-4	2	-2	29
2200 - 2299	-5	2	-3	28
2300 - 2399	-6	2	-4	27
2400 - 2499	-6	3	-3	28
2500 - 2599	-7	3	-4	27
2600 - 2899	-8	3	-5	26

7. Beregningen af Påskedag

Specielle forhold ved den gregorianske epakt

Ved beregningen af Påskedag kan der i den julianske kalender med den faste relation gyldental – epakt kun optræde 19 forskellige epakter, mens der i den gregorianske kalender, efterhånden som hundredårene går, vil være 30 forskellige epakter.

Derfor indførte den romersk-katolske kirke ved kalenderreformen i 1582 to mindre korrektioner, som dog kun vil være nødvendige i enkelte tilfælde [5],[6]:

- Påskedag kan også cyklisk indtræffe 26.april. Det foregår, når forårsjævndøgn 21.marts, fuldmåner 20.marts, 19.april samtidig med at det er søndag 19.april, men det må ikke finde sted med de fastlagte ydergrænser fra Alexandria, så her skal Påskedag være 19.april.
Det sker fx i 1981.
- Man kan i visse 100-års perioder komme ud for, at epakt 24 og 25 begge optræder. Så vil Påskedag for begge epakter altid være den samme indenfor den 19-årige cyklus, hvad der antyder at 19-års cyklussen er ufuldkommen. Det må ikke være indtrykket, så derfor skal man benytte dobbeltepakten (25) i stedet for epakt 25. Det giver ikke anledning til problemer, da epakt 26 ikke optræder samtidig med epakt 24 og 25. Med samme søndagsbogstaver giver dobbeltepakten (25) så i én kombination en anden Påskedag end epakt 25.
Det sker fx i 1954 og 2049.

Forholdet mellem gyldental og gregoriansk epakt udvikler sig da på denne måde, hvor ens 100års-intervaller er slået sammen. Månens spring ligger stadig mellem gyldental 19 og gyldental 1.

gyldental Årsinterval	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1582 - 1699	1	12	23	4	15	26	7	18	29	10	21	2	13	24	5	16	27	8	19
1700 - 1899	30	11	22	3	14	25	6	17	28	9	20	1	12	23	4	15	26	7	18
1900 - 2199	29	10	21	2	13	24	5	16	27	8	19	30	11	22	3	14	(25)	6	17
2200 - 2299	28	9	20	1	12	23	4	15	26	7	18	29	10	21	2	13	24	5	16
2300 - 2399	27	8	19	30	11	22	3	14	25	6	17	28	9	20	1	12	23	4	15
2400 - 2499	28	9	20	1	12	23	4	15	26	7	18	29	10	21	2	13	24	5	16
2500 - 2599	27	8	19	30	11	22	3	14	25	6	17	28	9	20	1	12	23	4	15
2600 - 2899	26	7	18	29	10	21	2	13	24	5	16	27	8	19	30	11	22	3	14

Og med disse korrektioner bliver den gregorianske påskedagstabel, som også er bearbejdet fra [5], således:

7. Beregningen af Påskedag

Tabel 7.6 Gregoriansk Påskedag											
Søndags- bogstav	Gregoriansk epakt									Påskedag	
D, ED	23										mar 22
	22	21	20	19	18	17	16				mar 29
	15	14	13	12	11	10	9				apr 5
	8	7	6	5	4	3	2				apr 12
	1	30	29	28	27	26 (25)	25	24			apr 19
E, FE	23	22									mar 23
	21	20	19	18	17	16	15				mar 30
	14	13	12	11	10	9	8				apr 6
	7	6	5	4	3	2	1				apr 13
	30	29	28	27	26 (25)	25	24				apr 20
F, GF	23	22	21								mar 24
	20	19	18	17	16	15	14				mar 31
	13	12	11	10	9	8	7				apr 7
	6	5	4	3	2	1	30				apr 14
	29	28	27	26 (25)	25	24					apr 21
G, AG	23	22	21	20							mar 25
	19	18	17	16	15	14	13				apr 1
	12	11	10	9	8	7	6				apr 8
	5	4	3	2	1	30	29				apr 15
	28	27	26 (25)	25	24						apr 22
A, BA	23	22	21	20	19						mar 26
	18	17	16	15	14	13	12				apr 2
	11	10	9	8	7	6	5				apr 9
	4	3	2	1	30	29	28				apr 16
	27	26 (25)	25	24							apr 23
B, CB	23	22	21	20	19	18					mar 27
	17	16	15	14	13	12	11				apr 3
	10	9	8	7	6	5	4				apr 10
	3	2	1	30	29	28	27				apr 17
	26 (25)	25	24								apr 24
C, DC	23	22	21	20	19	18	17				mar 28
	16	15	14	13	12	11	10				apr 4
	9	8	7	6	5	4	3				apr 11
	2	1	30	29	28	27	26 (25)				apr 18
	25	24									apr 25

Påskedagsberegning, delvis fælles for begge kalendere

Sammenligner man tabel 7.3 og 7.6, kan man se at udgangspunktet for den Alexandrinske og den gregorianske epakt er den samme, nemlig nymånens alder 1. januar. Derved bliver det muligt delvis at slå beregningerne sammen for de 2 kalendere. Grundidéen i algoritmen stammer fra [2], som nøje følger de angivne tabeller.

Når algoritmen skal gælde for Danmark-Norge, skal man lige huske at korrigere for år 1744, hvor den protestantiske beregning blev benyttet (og ligeledes en evt. korrektion for 1724 i hertugdømmerne).

Algoritme 11, PÅSKEDAG, Danmark og Norge

```

-----
/* P Å S K E D A G */

procedure påskedag(år,md,dg);
integer år,md,dg;
/* beregner påskedag, gælder for dansk-norsk kalender 600 – 3199 */

begin
integer ugedagpos, gyldental, hundredår, soljævning, månejævning,
epakt, pgrænse;

gyldental := år mod 19 + 1;
ugedagpos := (5 * år) // 4; /* ugedag for 1.marts, for år 0: mandag */

if år > 1699
then begin /* gregoriansk kalender */
  hundredår := år//100;
  soljævning := (3 * hundredår – 45) // 4;
  månejævning := (8 * hundredår – 112) // 25;
  ugedagpos := ugedagpos – 10 – soljævning ; /* korrigerer ugedag for 1.marts */
  epakt := (11 * gyldental + 19 + månejævning – soljævning) mod 30 + 1;

  /* korrigerer de 2 gregorianske undtagelser */
  if (epakt = 25 and gyldental > 11) or epakt = 24
  then epakt := epakt + 1;
end
else /* juliansk kalender */
  epakt := (11 * gyldental – 4) mod 30 + 1;

/* restberegningen er fælles for begge kalendere */
pgrænse := 44 – epakt;
if pgrænse < 21
then pgrænse := pgrænse + 30;
dg := pgrænse + 7 – (ugedagpos + pgrænse) mod 7;
if dg > 31
then begin
  md := 4;
  dg := dg – 31
end
else md := 3;

/* undtagelsen år 1744 for Danmark-Norge */
if år = 1744
then begin
  md := 3; dg := 29
end;

return md,dg;

end PÅSKEDAG
-----

```

8. Klassifikation af en dato

Efter beregningen af Påskedag er man nu i stand til at opstille en kalender for året, dvs. man kan klassificere alle datoerne alt efter om de er:

- almindelige hverdage (mandag – fredag)
- specielle hverdage (mandag – fredag med en speciel betydning eller placering)
- lørdage (almindelige og specielle)
- søndage (almindelige og specielle)
- skæve helligdage

Nationale helligdage

Fra og med dette kapitel begynder de nationale hellig- og mærkedage at spille en afgørende rolle, dvs. de følgende algoritmer er specifikt danske. Andre landes højtider må man – efter at have gennemgået de historiske fakta - udarbejde specielle algoritmer for.

Danske helligdage – en speciel situation

De danske helligdage er ret enestående sammenlignet med andre lande.

Siden Struensees reformer i 1770 [7], hvor han afskaffede en række ældre helligdage⁴, der havde overlevet Reformationen i 1536, har der siden 1771 kun været 2 ændringer mht. arbejds-/fridage:

- 1849 kom Grundlovsdagen, 5.juni, til
- 1891 kom 1.maj til

Klassifikation af en dato i Danmark

Man kan dog ikke fra kalenderen alene fastslå, om en forelagt dato er fri- eller arbejdsdag – dertil er de lokale forhold afgørende, men der kan opstilles en liste over alle de specielle dage, så brugerne **selv** kan tage stilling til, hvor de skal høre til.

I Danmark er det hensigtsmæssigt at klassificere en dato efter 4 grupper:

- Almindelig hverdag (mandag – fredag)
- Speciel hverdag (mandag – fredag), som enten har en speciel betydning, eller som grænser op til skæve helligdage. Derved skal de markeres.
- Lørdag, som ikke på forhånd er helligdag. Også her er der dage i forbindelse med højtider, der skal markeres specielt.
- Søn- og helligdage. Skæve helligdage og søndage i forbindelse med højtider skal markeres specielt.

⁴ Fra og med 1771 afskaffedes Hellig 3 Konger (6.januar), Kyndelmisse (2.februar), Marie Bebudelsesdag (25.marts), 3.Påskedag, 3.Pinsedag, St.Hansdag (24.juni), Marie Besøgelsesdag (2.juli), Mikkelsdag (29.september), Allehelgensdag (1.november), Mortensdag (11.november) og 3.Juledag. Store Bededag indført i 1686 blev ikke berørt.

8. Klassifikation af en dato

Med disse overvejelser er et forslag til klassifikation således:

Tabel 8.1 Klassifikation af en dato				
Dato	Gruppe	Ugedag	Beskrivelse	placering i forhold til Påskedag
Alle mandage-fredage, ikke nævnt nedenfor	0	varierer	0	varierer
Specielle mandage-fredage				
onsdag før Skærtorsdag	1	O	101	P - 4
tirsdag efter 2.Påskedag	1	Ti	102	P + 2
1.maj	1	varierer	103	fast dato
torsdag før Store Bededag	1	To	104	P + 25
onsdag før Kr.Himmelfartsdag	1	O	105	P + 38
fredag efter Kr.Himmelfartsdag	1	F	106	P + 40
Grundlovsdag, 5.juni	1	varierer	107	fast dato
fredag før Pinsedag	1	F	108	P + 47
tirsdag efter 2.Pinsedag	1	Ti	109	P + 51
Lillejuleaftensdag, 23.dec	1	varierer	110	fast dato
Juleaftensdag, 24.dec	1	varierer	111	fast dato
dagen efter 2.Juledag, 27.dec	1	varierer	112	fast dato
Nytårsaftensdag, 31.dec	1	varierer	113	fast dato
Lørdage				
lørdag før Palmesøndag	2	L	201	P - 8
Påskelørdag	2	L	202	P - 1
Store Bededags-lørdag	2	L	203	P + 27
lørdag efter Kr.Himmelfartsdag	2	L	204	P + 41
Pinselørdag	2	L	205	P + 48
alle andre lørdage	2	L	200	varierer
Søn- og helligdage				
Nytårsdag, 1.jan	3	varierer	301	fast dato
Fastelavn	3	S	302	P - 49
Palmesøndag	3	S	303	P - 7
Skærtorsdag	3	To	304	P - 3
Langfredag	3	F	305	P - 2
Påskedag	3	S	306	P
2.Påskedag	3	M	307	P + 1
Store Bededag	3	F	308	P + 26
Søndag efter Store Bededag	3	S	309	P + 28
Kr.Himmelfartsdag	3	To	310	P + 39
søndag efter Kr.Himmelfartsdag	3	S	311	P + 42
Pinsedag	3	S	312	P + 49
2.Pinsedag	3	M	313	P + 50
Juledag, 25. dec	3	varierer	314	fast dato
2.Juledag, 26. dec	3	varierer	315	fast dato
alle andre søndage	3	S	300	varierer

Der er lige en bemærkning til klassifikationen.

Hvis 1.maj og/eller Grundlovsdag, 5.juni, falder på en helligdag (gruppe 3), speciel lørdag (gruppe 2) eller en anden speciel mandag-fredag (gruppe 1), så ommarkeres de berørte datoer, da forfatteren her har **valgt** at prioritere klassifikationerne højere end standardklassifikationer i gruppe 1, (speciel mandag-fredag). Men aktuelt kan man ved omprogrammering selv vælge en anden prioritering.

Så kan algoritmen opstilles. Den er ikke særlig elegant i forhold til de tidligere, da der skal søges enten i forskydelige eller faste datoer.

Algoritme 12, DAGTYPE, Danmark

```

-----
/* D A G T Y P E */

procedure dagtype(år,md,dg, gruppe, ugedag, beskriv);
integer år, md, dg, gruppe, ugedag, beskriv;
/* klassificerer en dato, gælder for dansk kalender 1771 – 3199 */

begin
integer jtal, dagantal, pmd, pdg, i, mddg;
integer array reldg[1:24], grp[1:24], besk[1:24], fastdato[1:9], fastgrp[1:9], fastbesk[1:9];
boolean fundet;

/* placér de forskydelige dage i forhold til Påskedag */
reldg[1] := -49; grp[1] := 3; besk[1] := 302; /* Fastelavn */
reldg[2] := -8; grp[2] := 2; besk[2] := 201; /* lørdag før Palmesøndag */
reldg[3] := -7; grp[3] := 3; besk[3] := 303; /* Palmesøndag */
reldg[4] := -4; grp[4] := 1; besk[4] := 101; /* onsdag før Skærtorsdag */
reldg[5] := -3; grp[5] := 3; besk[5] := 304; /* Skærtorsdag */
reldg[6] := -2; grp[6] := 3; besk[6] := 305; /* Langfredag */
reldg[7] := -1; grp[7] := 2; besk[7] := 202; /* Påselørdag */
reldg[8] := 0; grp[8] := 3; besk[8] := 306; /* Påskedag */
reldg[9] := 1; grp[9] := 3; besk[9] := 307; /* 2.Påskedag */
reldg[10] := 2; grp[10] := 1; besk[10] := 102; /* tirsdag efter 2.Påskedag */
reldg[11] := 25; grp[11] := 1; besk[11] := 104; /* torsdag før Store Bededag */
reldg[12] := 26; grp[12] := 3; besk[12] := 308; /* Store Bededag */
reldg[13] := 27; grp[13] := 2; besk[13] := 203; /* Store Bededags-lørdag */
reldg[14] := 28; grp[14] := 3; besk[14] := 309; /* søndag efter Store Bededag */
reldg[15] := 38; grp[15] := 1; besk[15] := 105; /* onsdag før Kristi Himmelfartsdag */
reldg[16] := 39; grp[16] := 3; besk[16] := 310; /* Kristi Himmelfartsdag */
reldg[17] := 40; grp[17] := 1; besk[17] := 106; /* fredag efter Kristi Himmelfartsdag */
reldg[18] := 41; grp[18] := 2; besk[18] := 204; /* lørdag efter Kristi Himmelfartsdag */
reldg[19] := 42; grp[19] := 3; besk[19] := 311; /* søndag efter Kristi Himmelfartsdag */
reldg[20] := 47; grp[20] := 1; besk[20] := 108; /* fredag før Pinselørdag */
reldg[21] := 48; grp[21] := 2; besk[21] := 205; /* Pinselørdag */
reldg[22] := 49; grp[22] := 3; besk[22] := 312; /* Pinsedag */
reldg[23] := 50; grp[23] := 3; besk[23] := 313; /* 2.Pinsedag */
reldg[24] := 51; grp[24] := 1; besk[24] := 109; /* tirsdag efter 2. Pinsedag */

/* placér de faste datoer i forhold til 1.marts, for Nytårsdag 1.marts året før */
fastdato[1] := 101; fastgrp[1] := 3; fastbesk[1] := 301; /* Nytårsdag, 1.januar */
fastdato[2] := 501; fastgrp[2] := 1; fastbesk[2] := 103; /* 1.maj */
fastdato[3] := 605; fastgrp[3] := 1; fastbesk[3] := 107; /* Grundlovsdag, 5.juni */
fastdato[4] := 1223; fastgrp[4] := 1; fastbesk[4] := 110; /* Lillejuleaftensdag, 23.dec */
fastdato[5] := 1224; fastgrp[5] := 1; fastbesk[5] := 111; /* Juleaftensdag, 24.dec */
fastdato[6] := 1225; fastgrp[6] := 3; fastbesk[6] := 314; /* Juledag, 25.dec */
fastdato[7] := 1226; fastgrp[7] := 3; fastbesk[7] := 315; /* 2.Juledag, 26.dec */
fastdato[8] := 1227; fastgrp[8] := 1; fastbesk[8] := 112; /* dagen efter 2.Juledag, 27.dec */
fastdato[9] := 1231; fastgrp[9] := 1; fastbesk[9] := 113; /* Nytårsaftensdag, 31.dec */
/* korriger 1.maj og 5.juni, når de ikke skal markeres specielt */
if år < 1891
then fastgrp[2] := fastbesk[2] := 0;
if år < 1849
then fastgrp[3] := fastbesk[3] := 0;
/* find ugedagen for datoen - enten algoritme 8, uge eller algoritme 5, jdag */

```

8. Klassifikation af en dato

```
/* ENTEN */   uge(år,md,dg,ugenr,ugedag);
/* ELLER */   jtal := jdag(år, md, dg);
              ugedag := jtal mod 7 + 1;

/* beregn Påskedag, og find antal dage mellem datoen og den,
   algoritme 11, påskedag og algoritme 7, datodiff */
påskedag(år, pmd, pdg);
dagantal := datodiff(år, md, dg, år, pmd, pdg);

/* beregn datogruppe og datobeskrivelse */
gruppe := 0;
if ugedag > 5
then gruppe := ugedag - 4;
beskriv := gruppe * 100;

/* hvis datoen ligger inden for omfanget af de forskydelige dage,
   (1.februar - 14.juni, ydergrænserne for hhv. Fastelavn og 2.Pinsedag),
   skal der søges i tabellen over disse dage */
if dagantal > -50 and dagantal < 52
then begin
  /* bare primitiv søgning */
  fundet := false;
  i := 1;
  while i < 25 and not fundet do
  begin
    if dagantal = reldg[i]
    then begin
      gruppe := grp[i];
      beskriv := besk[i];
      fundet := true
    end;
    i := i + 1
  end while;
end;
/* evt. søgning i de forskydelige datoer er afsluttet,
   søg nu i tabellen over faste datoer, eventuelle lørdage og søndage
   vil overskrive mandage-fredage i gruppen */
/* bare primitiv søgning */
fundet := false;
i := 1;
while i < 10 and not fundet do
begin
  if mddg = fastdato[i]
  then begin
    gruppe := fastgrp[i];
    beskriv := fastbesk[i];
    fundet := true
  end;
  i := i + 1
end while;

return gruppe, ugedag, beskriv;

end DAGTYPE
```

Undersøgelse af et datointerval i Danmark

Det er også muligt at generalisere DAGTYPE, så man undersøger hvor mange dage af hver gruppe, der ligger i et forelagt datointerval, fx en måned, et kvartal eller et år.

Når antallene foreligger, kan man så lokalt afgøre antallet af arbejdsdage og fridage i det angivne tidsrum.

Den her forelagte algoritme ARBDIFF tæller lidt færre kategorier op end klassifikationerne i DAGTYPE, men den kan udvides, hvis man er interesseret i specielle datoer.

ARBDIFF kan også betragtes som en uddybning af algoritme 6, DATODIFF, og af praktiske årsager er størst tilladte tidsinterval 11 kalenderår. Den tidligste angivne dato tæller med, mens den seneste ikke gør det, fx vil ARBDIFF med kald af start og slutdato hhv. (2008,09,01) og (2008,10,01) optælle og sortere alle datoer i september 2008.

ARBDIFF tæller op i disse kategorier:

Tabel 8.2 Optalte kategorier af datoer	
antal[1] = antal[2] + antal[3] + antal[4] + antal[5] + antal[6]	
antal[6] = antal[7] + antal[8] + antal[9] + antal[10] + antal[11]	
antal[]	kategori
1	dage i alt
2	søndage, inkl. helligdage på søndage
3	lørdage, der er helligdage
4	almindelige lørdage
5	mandage-fredage, der er helligdage
6	mandage-fredage, der ikke er helligdage
7	1. maj, der er almindelige mandage-fredage
8	5. juni, der er almindelige mandage-fredage
9	24. dec, der er almindelige mandage-fredage
10	31. dec, der er almindelige mandage-fredage
11	alle resterende almindelige mandage-fredage

Algoritme 13, ARBDIFF, Danmark

/* A R B D I F F */

procedure arbdiff(år1, md1, dgl, år2, md2, dg2, antal);

integer år1,md1,dg1,år2,md2,dg2;

integer array antal[1:11];

/* beregner antallet af arbejdsdage mellem 2 datoer,
så helligdage og specielle hverdage også bliver talt op
tidligste dato tælles med, seneste ikke.

gælder for Danmark fra 1771

antal[1:11]

1 dage i alt

2 søndage, inklusive helligdage, der falder på søndage

3 lørdage, der er helligdage

4 lørdage, almindelige

5 mandage-fredage, der er helligdage

6 mandage-fredage, der ikke er helligdage

8. Klassifikation af en dato

- 7 maj 01, der er alm. mandage-fredage
- 8 Grundlovsdage, der er alm. mandage-fredage
- 9 Juleaftensdage, der er alm. mandage-fredage
- 10 Nytårsaftensdage, der er alm. mandage-fredage
- 11 resterende mandage-fredage, helt almindelige

Der gælder:

antal[1] = antal[2] + antal[3] + antal[4] + antal[5] + antal[6]

antal[6] = antal[7] + antal[8] + antal[9] + antal[10] + antal[11] */

begin

integer jdnr1, jdnr2, hjtal, ugedagkod1, ugedagkod2, årstop, årstal,

jbasisfør, jbasis, i, pmd, pdg, jdnrp;

integer array afst[1:13], jdato[1:13], ugedage[1:4];

/* basis for et år er mar 00, dvs. dagen før mar 01, i året */

/* afst[1:6] er afstande fra Påskedag i et år */

afst[1] := -3; /* Skærtorsdag */

afst[2] := -2; /* Langfredag */

afst[3] := 1; /* 2. Påskedag */

afst[4] := 26; /* St. Bededag */

afst[5] := 39; /* Kr.Himf.dag */

afst[6] := 50; /* 2.Pinsedag */

/* afst[7] er afstanden fra basis året før */

afst[7] := 307; /* Nytårsdag */

/* afst[8:13] er afstande fra basis i året til de andre faste datoer */

afst[8] := 62; /* maj 01 */

afst[9] := 97; /* Grundlovsdag */

afst[10] := 299; /* Juleaftensdag */

afst[11] := 300; /* Juledag */

afst[12] := 301; /* 2.Juledag */

afst[13] := 306; /* Nytårsaftensdag */

/* jdato[1:13] er de julianske dagtal for de 13 datoer */

/* ugedage[1:4] er ugedage for de 7 faste datoer, de falder i 4 grupper

1. jan 01,

2. dec 24, dec 31

3. maj 01, jun 05, dec 25

4. dec 26 */

/* nulstil optællingsdata */

for i := 1 **step** 1 **until** 13 **do** antal[i] := 0;

/* beregn juliansk dagtal for tidligste og seneste dato, algoritme 5, jdag */

jdnr1 := jdag(år1,md1,dg1);

jdnr2 := jdag(år2,md2,dg2);

/* byt evt. om, så tidligste dato altid er nr 1 */

if jdnr2 < jdnr1

then begin

hjtal := jdnr1; jdnr1 := jdnr2; jdnr2 := hjtal;

år1 := år2; år2 := år1; år1 := hjtal;

md1 := md2; md2 := md1; md1 := hjtal;

dg1 := dg2; dg2 := dg1; dg1 := hjtal

end;

8. Klassifikation af en dato

```
/* årstop anvendes til at styre årsskifter */
årstop := år2;
if md2 = 1 and dg2 = 1
then årstop := år2 - 1;

/* kontrollér intervallet, max 11 kalenderår = 4016 til 4018 dage */
if årstop - år1 > 10
then quit;                                /* fejlreaktion */

/* beregn først dagantal og totalt antal lørdage og søndage */
antal[1] := jdnr2 - jdnr1;                    /* kunne også have brugt algoritme 7, datodiff */
ugedagkodl := jdnr1 mod 7;
ugedagkod2 := jdnr2 mod 7;
antal[4] := (antal[1] + (ugedagkodl + 1) mod 7) // 7;
antal[2] := (antal[1] + ugedagkodl) // 7;

/* de resterende beregninger skal styres af årstallene, opsamling af antal sker inden for ét årstal af gangen */
årstal := år1;

/* beregn juliansk dagtal for basis året før, algoritme 5, jdag */
jbasisfør := jdag(årstal - 1,3,1) - 1;

/* behandling af et enkelt år, som intervallet omfatter helt eller delvis */
while årstal <= årstop do
begin
  /* juliansk dagtal for basis i året */
  jbasis := jdag(årstal,3,1) - 1;

  /* juliansk dagtal og ugedage for de 7 faste datoer i året:
  jan 01, maj 01, jun 05, dec 24, dec 25, dec 26, dec 31 */

  jdato[7] := jbasisfør + afst[7];
  for i := 8 step 1 until 13 do
    jdato[i] := jbasis + afst[i];
  ugedage[1] := jdato[7] mod 7;
  ugedage[2] := jdato[10] mod 7;
  ugedage[3] := jdato[8] mod 7;
  ugedage[4] := jdato[12] mod 7;

  /* beregn Påskedag i året, og juliansk dagtal for den, algoritmerne 5 og 11 */
  påskedag(årstal, pmd, pdg);
  jdnrp := jdag(årstal, pmd, pdg);

  /* beregn juliansk dagtal for de 6 forskydelige helligdage,
  og tæl dem op, hvis de hører med i det totale interval */
  for i := 1 step 1 until 6 do
  begin
    jdato[i] := jdnrp + afst[i];
    if jdnr1 <= jdato[i] and jdato[i] < jdnr2
      then antal[5] := antal[5] + 1;
  end;

  /* undersøg de 7 faste dage i året, og tæl dem op, hvis de hører med */
```

8. Klassifikation af en dato

```
/* jan 01, Nytårsdag */
if jdnr1 <= jdato[7] and jdato[7] < jdnr2
then begin
  if ugedage[1] < 5
  then antal[5] := antal[5] + 1
  else begin
    if ugedage[1] = 5
    then begin
      antal[4] := antal[4] - 1;
      antal[3] := antal[3] + 1;
    end
  end
end;

/* maj 01 */
if jdnr1 <= jdato[8] and jdato[8] < jdnr2
then begin
  if ugedage[3] < 5
  then begin
    /* undtag, hvis det er Store Bededag eller Kr.Himf.dag */
    if jdato[8] ≠ jdato[4] and jdato[8] ≠ jdato[5]
    then begin
      if årstal > 1890
      then antal[7] := antal[7] + 1
    end
  end
end;

/* jun 05, Grundlovsdag */
if jdnr1 <= jdato[9] and jdato[9] < jdnr2
then begin
  if ugedage[3] < 5
  then begin
    /* undtag, hvis 2.Pinsedag */
    if jdato[9] ≠ jdato[6]
    then begin
      if årstal > 1848
      then antal[8] := antal[8] + 1
    end
  end
end;

/* dec 24, Juleaftensdag */
if jdnr1 <= jdato[10] and jdato[10] < jdnr2
then begin
  if ugedage[2] < 5
  then antal[9] := antal[9] + 1
end;
```

8. Klassifikation af en dato

```
/* dec 25, Juledag */
if jdnr1 <= jdato[11] and jdato[11] < jdnr2
then begin
  if ugedage[3] < 5
  then antal[5] := antal[5] + 1
  else begin
    if ugedage[3] = 5
    then begin
      antal[4] := antal[4] - 1;
      antal[3] := antal[3] + 1;
    end
  end
end;

/* dec 26, 2. Juledag */
if jdnr1 <= jdato[12] and jdato[12] < jdnr2
then begin
  if ugedage[4] < 5
  then antal[5] := antal[5] + 1
  else begin
    if ugedage[4] = 5
    then begin
      antal[4] := antal[4] - 1;
      antal[3] := antal[3] + 1
    end
  end
end;

/* dec 31, Nytårsaftensdag */
if jdnr1 <= jdato[13] and jdato[13] < jdnr2
then begin
  if ugedage[2] < 5
  then antal[10] := antal[10] + 1
end;

/* flyt juliensk dagtal for basis til året før */
jbasisfør := jbasis;

årstal := årstal + 1
end while;

/* slut på undersøgelsen år for år,
beregnet til sidst totaltallene fra de opsamlede tal */

antal[6] := antal[1] - antal[2] - antal[3] - antal[4] - antal[5];
antal[11] := antal[6] - antal[7] - antal[8] - antal[9] - antal[10];

return antal[ ];

end ARBDIFF
-----
```

8. Klassifikation af en dato

9. Eksempler på datoberegninger

Dette kapitel indeholder en række eksempler på, hvad forfatteren har skullet løse. Så kan man få en fornemmelse af, hvad de forelagte algoritmer kan benyttes til.

Eksempel 1: Find datoen for evt. søndag mellem jul og nytår

Det er ikke alle år, der har en søndag fra og med 27.dec til og med 31.dec.
Givet: år. Find: søndag (månedens altid = 12).

```
-----
/* beregn ugedag for 27.dec, ENTEN algoritme 9: uge, ELLER algoritme 5: jdag */
/* ENTEN */   uge(år,12,27,ugenr,ugedagkod);
/* ELLER */   ugedagkod := jdag(år,12,27) mod 7 + 1;

/* ugedagkod=1: mandag, ..., =7: søndag */

/* hvis 27.dec er mandag eller tirsdag, er der ingen søndag overhovedet */
if ugedagkod < 3
then begin
  /* ingen søndag, fx kan man kode */
  søndag := 0
end
else søndag := 34 - ugedagkod;
-----
```

Eksempel 2: Find 1. torsdag i en måned

Givet: år, md. Find: dg.

```
-----
/* beregn ugedag for 1. i måneden, ENTEN algoritme 9: uge, ELLER algoritme 5: jdag */
/* ENTEN */   uge(år,md,1,ugenr,ugedagkod);
/* ELLER */   ugedagkod := jdag(år,md,1) mod 7 + 1;

/* ugedagkod=1: mandag, ..., =7: søndag */

if ugedagkod < 5
then begin
  /* 1. er mandag-torsdag */
  dg := 5 - ugedagkod
end
else begin
  /* 1. er fredag-søndag */
  dg := 12 - ugedagkod
end;
-----
```

Eksempel 3: Find 1. fredag efter 3. tirsdag i en måned

Givet: år, md. Find: nydg.

```
-----
/* beregn ugedag for 1. i måneden, ENTEN algoritme 9: uge, ELLER algoritme 5: jdag */
/* ENTEN */    uge(år,md,1,ugenr,ugedagkod);
/* ELLER */    ugedagkod := jdag(år,md,1) mod 7 + 1;

/* ugedagkod=1: mandag, ..., =7: søndag */

/* find 1. tirsdag (jfr. eksempel 2) */

if ugedagkod < 3
then begin
  /* 1. er mandag-tirsdag */
  ftirsdg := 3 - ugedagkod
end
else begin
  /* 1. er onsdag-søndag */
  ftirsdg := 10 - ugedagkod
end;

/* 3.fredag er 14 + 3 dage efter 1.tirsdag,
   ENTEN algoritme 8: nydato, ELLER direkte hvis man er sikker (!) på, at måneden har mindst 24 dage */
/* ENTEN */    nydato(år,md,ftirsdg,17,nyår,nymd,nydg);
/* ELLER */    nydg := ftirsdg + 17;
-----
```

Eksempel 4: Find startdato for en uge, der indeholder en given dato

Givet: år, md, dg. Find: nyår, nymd, nydg.

```
-----
/* man kan gå 2 veje, ENTEN ved at benytte ugenummeret, ELLER ved at tælle dage med nydato */

/* 1. metode: gennem ugenummeret, algoritme 9: uge og algoritme 10: ugestart */

/* ugenummer og ugedag for den givne dato */
uge(år,md,dg,ugenr,ugedagkod);

/* dato for 1.dag i ugen */
ugestart(år,ugenr,1,nyår,nymd,nydg);

/* 2.metode: gennem algoritme 6: jdag og algoritme 8: nydato */

/* ugedag for den givne dato */
ugedagkod := jdag(år,md,dg) mod 7 + 1;

/* find startdato (mandag) i ugen */
nydato(år,md,dg,1-ugedagkod,nyår,nymd,nydg);
-----
```

Eksempel 5: Beregning af forfaldsdato (bankdag)

Et betalingsproblem:

"Forfaldsdato (dvs. sidste dato for korrekt betaling) for en måned er sidste dag i måneden. Hvis datoen er en lørdag, søndag eller helligdag, er det førstkommande hverdag derefter".

Her skal man først have afklaret, **hvilke** dage, der henregnes til 'lørdag, søndag eller helligdag'.

I de seneste år er man i Danmark gået over til at definere hverdage som **bankdage**, dvs. dage, hvor betalinger kan registreres. Ikke-bankdage ('helligdage') er af Finansrådet fastlagt som: Lørdage, søndage, helligdage, Grundlovsdag (5.juni) og Juleaftensdag (24.dec). Fra og med år 2003 er Nytårsaftensdag (31.dec) heller ikke mere bankdag.

Overvejelser af denne slags er typiske for datoberegninger, og det anbefales kraftigt, at man **tager højde for alle typer af datoer**. Man støder på de overvejelser den ene gang efter den anden, så det efterhånden er blevet til en rygmarvsreaktion ved alle beregninger.

I dette eksempel blev problemet ved ihærdige forespørgsler og 'boren i emnet' kogt ned til bankdag / ikke-bankdag, dvs.:

"Forfaldsdato for en måned er sidste bankdag i måneden. Hvis datoen er en ikke-bankdag, er det førstkommande bankdag derefter. Beregn den for en given måned".

Koden er kun skitseret, Givet: år, md. Find: forfår, forfmd, forfdg.

```
-----
/* find næste måned – og korrekt år */
if md = 12
then begin
  næstår := år + 1;
  næstmd := 1
end
else begin
  næstår := år;
  næstmd := md + 1
end;

/* find sidste dag i givne måned, ENTEN algoritme 7, datodiff, ELLER algoritme 8, nydato */
/* ENTEN */   sidstedg := datodiff(næstår,næstmd,1,år,md,1); /* sidste dag er også antallet i måneden */
/* ELLER */   nydato(næstår,næstmd,1,-1,år,md,sidstedg); /* 1 dag før den 1. i næste måned */

/* undersøg så dagtype for sidste dag i måneden, algoritme 12 */
dagtype(år,md,sidstedg,gruppe,ugedag,beskriv);

/* resten af programmet kan med fordel tabelstyres */
/* for gruppe = 0 (almindelig hverdag) er den sidste dag også bankdag */

/* for gruppe = 1 (speciel hverdag) med beskriv ≠ (5.jun og 24. dec og 31.dec fra 2003)
er sidste dag også bankdag.
Med beskriv = 5.jun eller 24.dec og 31.dec kan man med nydato og ugedag regne sig frem til 1.bankdag */

/* for gruppe = 2 (lørdag) kan man af beskriv regne sig frem til 1.bankdag med nydato */

/* for gruppe = 3 (søn- og helligdag) kan man af beskriv ligeledes regne sig frem til
1.bankdag med nydato */
-----
```

9. Eksempler på datoberegninger

10. Beregning af tidspunkter

Underinddelinger af døgn

Når man kan regne i antal hele dage (døgn), er det let at udvide beregningerne til underinddelinger af døgn, så man også kan operere med timer, minutter og sekunder.

Krav til nøjagtighed af beregningerne

Indtil nu har man kunnet nøjes med evt. interkalationer af dage, men så snart man vil gå længere i nøjagtighed, er man nødt til at tage højde for flere besværlige forhold:

- indsættelse af skudsekunder på grund af Jordens ikke helt regelmæssige rotation
- overgange til og fra sommertid i forskellige lande

Skudsekunder – indførelse af atomtid

I det følgende forudsættes det, at tidsmålingen er **TAI, Atomtiden***, som altid er regnet til 0°-meridianen. Tidligere blev den betegnet som **ET, Efemeridetid***. TAI er strengt ensartet uden justeringer, hvorimod vores almindelige klokkeslæt af og til må have indsat **skudsekunder*** for at have et tilnærmet antal hele sekunder i forskel mellem TAI og vores almindelige **universaltid***, **UT** eller **UTC**, **Greenwich tid**, der er uregelmæssig og følger Jordens rotation. Indsættelserne sker bagud efter målinger af afvigelse.

Ved nøjagtig beregning af sol- og måneformørkelser er man nødt til at anvende TAI, dvs. beregne (eller historisk anslå) $\Delta T = ET - UT$. Senere er der indført flere tidsbetegnelser, der afviger en lille smule fra UTC og som tager hensyn til relativistiske effekter.

En mere udførlig redegørelse for de forskellige tidsmålinger findes i [25].

Tabel 10.1 indeholder de målte forskelle mellem TAI og UTC fra [21],[24] og [25], hvor der refereres til dem som hhv. ET og UT.

år	sek	år	sek	år	sek	år	sek	år	sek
1681	-14	1852	3	1908	8	1942	25	1976	47
1710	-12	1857	3	1910	10	1944	26	1978	49
1727	-7	1862	3	1912	13	1946	27	1980	51
1737	-3	1867	1	1914	15	1948	28	1982	53
1747	-1	1872	-5	1916	17	1950	29	1984	55
1755	1	1877	-7	1918	19	1952	30	1986	55
1771	6	1882	-8	1920	20	1954	31	1988	56
1785	8	1887	-8	1922	22	1956	32	1990	57
1792	8	1890	-7	1924	22	1958	32	1992	59
1801	6	1892	-8	1926	23	1960	33	1994	61
1809	5	1894	-8	1928	23	1962	34	1996	62
1813	5	1896	-7	1930	23	1964	35	1998	64
1822	4	1898	-6	1932	24	1966	37	2000	65
1831	1	1900	-4	1934	24	1968	39	2002	65
1837	0	1902	-1	1936	24	1970	41	2004	65
1843	1	1904	1	1938	24	1972	43		
1848	2	1906	5	1940	24	1974	45		

Til historiske beregninger, uden fluktuationer, anvender man traditionelt formelen [21],[24]:

$$\Delta T = +24,349 + 72,318 T + 29,950 T^2 \quad (60)$$

hvor T måles i julianske århundreder på 36525 atomdøgn (efemeridedøgn) fra JD 2415020,0 = 1899 dec 31 kl 12:00 TAI (ET).

10. Beregning af tidspunkter

Sommertid

Ved tidsberegningerne er der heller ikke taget hensyn ved skift til og fra sommertid, hvor urene typisk stilles 1 time frem og tilbage for at udnytte den lyse tid mest effektivt.

Tabel 10.2 indeholder en liste over sommertid i Danmark til og med 2005 [26],[27].

fra		til		fra		til	
dato	kl	dato	kl	dato	kl	dato	kl
1916-05-15		1916-09-30		1989-03-26	02:00	1989-09-24	03:00
1940-05-15	00:00	1942-11-02	03:00	1990-03-25	02:00	1990-09-30	03:00
1943-03-29	02:00	1943-10-04	03:00	1991-03-31	02:00	1991-09-29	03:00
1944-04-03	02:00	1944-10-02	03:00	1992-03-29	02:00	1992-09-27	03:00
1945-04-02	02:00	1945-08-15	03:00	1993-03-28	02:00	1993-09-26	03:00
1946-05-01	02:00	1946-09-01	03:00	1994-03-27	02:00	1994-09-25	03:00
1947-05-04	02:00	1947-08-10	03:00	1995-03-26	02:00	1995-09-24	03:00
1948-05-09	02:00	1948-08-08	03:00	1996-03-31	02:00	1996-10-27	03:00
1980-04-06	02:00	1980-09-28	03:00	1997-03-30	02:00	1997-10-26	03:00
1981-03-29	02:00	1981-09-27	03:00	1998-03-29	02:00	1998-10-25	03:00
1982-03-27	02:00	1982-09-25	03:00	1999-03-28	02:00	1999-10-31	03:00
1983-03-26	02:00	1983-09-24	03:00	2000-03-26	02:00	2000-10-29	03:00
1984-03-25	02:00	1984-09-30	03:00	2001-03-25	02:00	2001-10-28	03:00
1985-03-31	02:00	1985-09-29	03:00	2002-03-24	02:00	2002-10-27	03:00
1986-03-30	02:00	1986-09-28	03:00	2003-03-30	02:00	2003-10-26	03:00
1987-03-29	02:00	1987-09-27	03:00	2004-03-28	02:00	2004-10-31	03:00
1988-03-27	02:00	1988-09-25	03:00	2005-03-27	02:00	2005-10-30	03:00

Fra-klokkeslættet angiver det tidspunkt, hvor urene stilles 1 time frem, og til-klokkeslættet det tidspunkt, hvor urene stilles 1 time tilbage. Om foråret ”mangler” der således 1 time, og om efteråret optræder samme klokkeslæt indenfor en time 2 gange, så det er meget vigtigt at angive, **hvilken tid** klokkeslæt refererer til for at få en helt korrekt tidsberegning.

Kun sikre angivelser for fortiden

Tabel 10.1 og 10.2 indeholder en stor begrænsning i forhold til tabellerne i de tidligere kapitler – her er der i sagens natur kun - nogenlunde - sikre **angivelser for fortiden**. Hvad fremtiden vil bringe, er man nødt til at holde sig løbende orienteret om. Således er oplysningerne om sommertid 2005 taget fra en tidlig uddelt vægkalender.

Samme teknik som ved dagberegninger – Storbritanniens varianter af algoritmer

Først en algoritme til beregning af ”Juliansk atomtidspunkt” med given (kalender), år, måned, dag, TAI-time, TAI-minut, TAI-sekund, og derefter som ved dagberegningerne. Man skal lige huske på, at nyt dagtal begynder kl 12:00 TAI.

Algoritmerne gælder for Storbritannien med TAI, og derfor skal man anvende deres varianter af algoritmerne 5 og 6 , jdag og dato_j. Skiftet mellem juliansk og gregoriansk kalender foregik i 1752, hvor der blev sprunget fra sep 2 til sep 14.

Udpakning af decimaldele af døgn til timer, minutter og sekunder

Ved omsætningerne har det vist sig mest hensigtsmæssigt først at have en algoritme, der kan udpakke decimaldele af et døgn til hele timer, minutter og sekunder med decimaldele. Derved er man mere frit stillet overfor om man vil have slutresultaterne i decimaldele af døgn eller i timer, minutter og sekunder.

Ved at gemme fortegnet for decimaldelen af et døgn kan man internt regne med ikke-negative tal, og derved kan de kendte operatorer benyttes ved udpakningen.

Algoritme 14, UDKDGN, generel

```

-----
procedure udpakdgn(døgndel,tim,min,sek);
integer tim,min;
real døgndel,sek;

/* udpakker decimaldele af et døgn til timer, minutter og sekunder,
   gælder generelt */

begin
integer fortegn;
real resttid;

/* gem fortegn, og regn numerisk */
fortegn := sign(døgndel);
if fortegn = -1
then døgndel := 0 - døgndel;

resttid := døgndel * 24;           /* resttid i timer med decimaler */
tim := entier(resttid);
resttid := (resttid - tim) * 60;   /* resttid indenfor 1 time i minutter med decimaler */
min := entier(resttid);
sek := (resttid-min) * 60;        /* resttid indenfor 1 minut i sekunder */

if fortegn = -1
then begin
  tim := 0 - tim; min := 0 - min; sek := 0 - sek
end;

return tim,min,sek;

end UDKDGN
-----

```

Algoritme 15, JTID, generel

```

-----
real procedure jtid(år,md,dg,tim,min,sek);
integer år,md,dg,tim,min;
real sek;

/* beregner et juliansk atomtidspunkt */
begin
jtid := jdag(år,md,dg) - 0,5           /* algoritme 5, Storbritanniens variant skal anvendes */
  + (tim + (min + sek/60) /60) /24;    /* indpak til decimaldele af døgn */

return jtid;

end JTID
-----

```

Algoritme 16, TIDJ, generel

```

-----
procedure tidj(jtid,år,md,dg,tim,min,sek);
integer jdag,år,md,dg,tim,min;
real jtid, sek;

/* beregner dato,tim,min,sek for et juliansk atomtidspunkt */
begin
real resttid;
jdag := entier(jtid);
resttid := jtid - jdag;

/* korrektion af det halve døgn i jtid før beregning af korrekt dato */
if resttid >= 0,5
then begin
    jdag := jdag + 1;
    resttid := resttid - 0,5
end
else resttid := resttid + 0,5;

datoj(jdag, år, md, dg);                /* algoritme 6, Storbritanniens variant skal benyttes */

udpakdgn(resttid,tim,min,sek);          /* algoritme 14 */

return år, md, dg, tim, min, sek;

end TIDJ
-----

```

Algoritme 17, TIDDIFF, generel

```

-----
real procedure tiddiff(år1,md1,dg1,tim1,min1,sek1, år2,md2,dg2,tim2,min2,sek2);
integer år1,md1,dg1,tim1,min1, år2,md2,dg2,tim2,min2;
real sek1, sek2;

/* beregner tidsforskellen i døgn med decimaler mellem 2 atomtidspunkter */

begin

tiddiff := jtid(år2,md2,dg2,tim2,min2,sek2) - jtid(år1,md1,dg1,tim1,min1,sek1);

/* man kan evt. arbejde videre ved numerisk at udpakke decimaldelen af døgn
   til timer, minutter og sekunder */

return tiddiff;

end TIDDIFF
-----

```


Algoritme 18, NYTID, generel

```

-----
procedure nytid(år,md,dg,tim,min,sek, dagdiff,timdiff,mindiff,sekdiff,
  nyår,nymd,nydg,nytim,nymin,nysek);
integer år,md,dg,tim,min, dagdiff,timdiff,mindiff,
  nyår,nymd,nydg,nytim,nymin;
real sek, sekdiff, nysek;

/* beregner et nyt tidspunkt i atomtid udfra et givent og et tidsrum i dage, tim, min og sek */

begin
real jtid1,jtid2,diff;

jtid1 := jtid(år,md,dg,tim,min,sek);
diff := dagdiff + (timdiff + (mindiff + sekdiff/60) /60) /24;
jtid2 := jtid1 + diff;
tidj(jtid2,nyår,nymd,nydg,nytim,nymin,nysek);          /* benytter algoritme 14, udpakdgn */

return nyår,nymd,nydg,nytim,nymin,nysek;

end NYTID
-----

```

10. Beregning af tidspunkter

11. Eksempler på algoritmer for andre lande

Nationale forhold - historie og politik har stor indflydelse

Med de forskellige staters historie kommer man ikke udenom disse forhold, når man skal undersøge deres kalendere i detaljer. Tag fx et parallelt - og grelt - eksempel: Gader i det tidligere DDR kan i dag have skiftet navn 5-6 gange siden 1918. (Kejsertid, Weimarrepublik, Naziregime, Stalinregime, afstalinisering og genforening). Hvordan ser det så ikke ud for helligdage i de enkelte stater?

Et helt aktuelt eksempel: Sverige vil fra år 2005 afskaffe 2.Pinsedag som helligdag og erstatte den med nationaldagen 6.juni.

Nationale varianter af algoritme 1 og 2

Dette kapitel indeholder derfor først eksempler på de nationale varianter af algoritme 1 og 2 (årtype og validato). Med de 2 algoritmer kan man gå videre, **hvis** man vel at mærke kender de nationale helligdage og deres historie.

Da de nationale udgaver i sagens natur ligner hinanden meget, kan man overveje at udforme dem generelt med overgangså, -md, og -dag som parametre, men for overskuelighedens skyld vises de som separate algoritmer.

Generel udgave af algoritme 5 og 6

Man kan godt skrive mere generelle udgaver af algoritmerne. Her vises eksempler på algoritme 5 og 6 (jdag og dato), hvor man med en boolesk variabel angiver, **hvilken** af de 2 kalendere man omsætter til og fra, mens man holder alle former for overgangsdatoer ude af billedet.

Disse mere generelle udgaver skal dog udbygges, når der er mere end 1 skift (fx Sverige, Frankrig).

Overgangsdatoer mellem kalenderne kan ses i Tillæg E for de forskellige stater og landområder, og med en generel udformning kan man også benytte de generelle udgaver til direkte omsætning mellem juliansk og gregoriansk dato.

Algoritme 1S, ÅRTYPE, Sverige

```
-----
integer procedure årtype(år);
integer år;
```

```
/* Sverige:
```

```
undersøger, om et år er et skudår eller et af overgangsårene (1700,1712,1753)
mellem juliansk og gregoriansk kalender.
man udelod 1 dag ved at springe fra 1700 feb 28 til 1700 mar 01.
man indskød 1 dag ved at tilføje 1712 feb 30.
man udelod 11 dage ved at springe fra 1753 feb 17 til 1753 mar 01.
```

```
årtype  typen for årstallet
1      almindeligt år (365 dage)
2      skudår (366 dage)
3      år 1700 (365 dage)
4      år 1712 (367 dage)
5      år 1753 (354 dage) */
```

```
begin
```

```
if år > 1753
```

```
then begin
```

```
/* gregoriansk kalender */
```

11. Eksempler på algoritmer for andre lande

```
if (år mod 4) ≠ 0
then årtype := 1
else begin
  if (år mod 100) ≠ 0
  then årtype := 2
  else begin
    if (år mod 400) ≠ 0
    then årtype := 1
    else årtype := 2
  end
end
end
else begin
  if not (år = 1700 or år = 1712 or år = 1753) /* juliansk og egen kalender */
  then begin
    if (år mod 4) ≠ 0
    then årtype := 1
    else årtype := 2
  end
  else begin
    if år = 1700 /* 1. overgangsår */
    then årtype :=3;
    if år = 1712 /* 2. overgangsår */
    then årtype :=4;
    if år = 1753 /* 3. overgangsår */
    then årtype :=5;
  end
end;

return årtype;

end ÅRTYPE, Sverige
```

Algoritme 2S, VALIDATO, Sverige

```
integer procedure validato(år,md,dg);
integer år,md,dg;
```

/* Sverige

kontrol af en dato fra år 600 - 3199.

overgangsårene mellem juliansk og gregoriansk kalender er år 1700, 1712, 1753.

man udelod 1 dag ved at springe fra 1700 feb 28 til 1700 mar 01.

man indskød 1 dag ved at tilføje 1712 feb 30.

man udelod 11 dage ved at springe fra 1753 feb 17 til 1753 mar 01.

validato:

0 ok, datoen er valid

1 fejl i dg

2 fejl i md

4 fejl i år

summer af (1,2,4) kan også forekomme */

11. Eksempler på algoritmer for andre lande

```
begin
validato := 0;

if år < 600 or år > 3199
then validato := validato+4;
if md < 1 or md > 12
then validato := validato+2;

/* skudår og overgangsår undersøges senere */
if dg < 1
then validato := validato+1
else begin
  if md = 2 or md = 4 or md = 6 or md = 9 or md = 11
  then begin
    if dg > 30
    then validato := validato+1;
  end
  else begin
    if dg > 31
    then validato := validato+1;
  end
end;

if validato > 0
then return validato;

/* kontrol af skudår og overgangsår */
if md = 2
then begin
  if årtype(år) = 1 and dg > 28
  then validato := validato+1;
  if årtype(år) = 2 and dg > 29
  then validato := validato+1;
  if årtype(år) = 3 and dg > 28
  then validato := validato+1;
  if årtype(år) = 5 and dg > 17
  then validato := validato+1;
end;

return validato;

end VALIDATO, Sverige
-----
```

Algoritme 1R, ÅRTYPE, Romersk-katolske kirke

```

-----
integer procedure årtype(år);
integer år;

/* Romersk-katolske kirke, Spanien, Portugal, Italienske fyrstendømmer
og i princippet kolonier, som dengang hørte til landene.
undersøger, om et år er et skudår eller overgangsåret 1582 mellem
juliansk og gregoriansk kalender.
man udelod 10 dage ved at springe fra 1582 okt 04 til 1582 okt 15.

årtype  typen for årstallet
1      almindeligt år(365 dage)
2      skudår(366 dage)
3      år 1582(355 dage) */

begin
if år > 1582                                /* gregoriansk kalender */
then begin
  if (år mod 4)  $\neq$  0
  then årtype := 1
  else begin
    if (år mod 100)  $\neq$  0
    then årtype := 2
    else begin
      if (år mod 400)  $\neq$  0
      then årtype := 1
      else årtype := 2
    end
  end
end
else begin
  if år < 1582                                /* juliansk kalender */
  then begin
    if (år mod 4)  $\neq$  0
    then årtype := 1
    else årtype := 2
  end
  else årtype := 3                                /* overgangsåret 1582 */
end;

return årtype;

end ÅRTYPE, Romersk-katolske kirke
-----

```

Algoritme 2R: VALIDATO, Romersk-katolske kirke

```

-----
integer procedure validato(år,md,dg);
integer år,md,dg;
/* Romersk-katolske kirke, Spanien, Portugal, Italienske fyrstendømmer
   og i princippet kolonier, som dengang hørte til landene.
   undersøger, om et år er et skudår eller overgangsåret 1582 mellem
   juliansk og gregoriansk kalender.
   man udelod 10 dage ved at springe fra 1582 okt 04 til 1582 okt 15.
validato
0 ok, datoen er valid
1 fejl i dg
2 fejl i md
4 fejl i år
summer af (1,2,4) kan også forekomme */

begin
validato := 0;
if år < 600 or år > 3199
then validato := validato+4;
if md < 1 or md > 12
then validato := validato+2;

/* skudår og overgangsår undersøges senere */
if dg < 1
then validato := validato + 1
else begin
  if md = 4 or md = 6 or md = 9 or md = 11
  then begin
    if dg > 30
    then validato := validato + 1;
  end
  else begin
    if md = 2 and dg > 29
    then validato := validato + 1;
    else begin
      if dg > 31
      then validato := validato + 1;
    end
  end
end;

if validato > 0
then return validato;
/* kontrol af skudår og overgangsår */
if årtype(år) ≠ 2 and md = 2 and dg = 29
then validato := validato+1;
if årtype(år) = 3 and md = 10 and dg > 4 and dg < 15
then validato := validato+1;

return validato;

end VALIDATO, Romersk-katolske kirke
-----

```

Algoritme 1U, ÅRTYPE, Storbritannien

```

-----
integer procedure årtype(år);
integer år;

/* Storbritannien samt kolonier, som dengang hørte til landet.
   undersøger, om et år er et skudår eller overgangsåret 1752 mellem
   juliansk og gregoriansk kalender.
   man udelod 11 dage ved at springe fra 1752 sep 02 til 1752 sep 14.
   årtype  typen for årstallet
   1      almindeligt år(365 dage)
   2      skudår(366 dage)
   3      år 1752(355 dage)   */

begin
if år > 1752                                /* gregoriansk kalender */
then begin
  if (år mod 4)  $\neq$  0
  then årtype := 1
  else begin
    if (år mod 100)  $\neq$  0
    then årtype := 2
    else begin
      if (år mod 400)  $\neq$  0
      then årtype := 1
      else årtype := 2
    end
  end
end
else begin
  if år < 1752                                /* juliansk kalender */
  then begin
    if (år mod 4)  $\neq$  0
    then årtype := 1
    else årtype := 2
  end
  else årtype := 3                                /* overgangsåret 1752 */
end;

return årtype;

end ÅRTYPE, Storbritannien
-----

```


Algoritme 2U: VALIDATO, Storbritannien

```

-----
integer procedure validato(år,md,dg);
integer år,md,dg;
/* Storbritannien samt kolonier, som dengang hørte til landet.
   kontrol af en dato fra år 600 - 3199.
   man udelod 11 dage ved at springe fra 1752 sep 02 til 1752 sep 14.
validato
0 ok, datoen er valid
1 fejl i dg
2 fejl i md
4 fejl i år
summer af (1,2,4) kan også forekomme */

begin
validato := 0;

if år < 600 or år > 3199
then validato := validato+4;
if md < 1 or md > 12
then validato := validato+2;

/* skudår og overgangsår undersøges senere */
if dg < 1
then validato := validato+1
else begin
  if md = 4 or md = 6 or md = 9 or md = 11
  then begin
    if dg > 30
    then validato := validato+1;
  end
  else begin
    if md = 2 and dg > 29
    then validato := validato+1;
    else begin
      if dg > 31
      then validato := validato+1;
    end
  end
end;

if validato > 0
then return validato;

/* kontrol af skudår og overgangsår */
if årtype(år) = 1 and md = 2 and dg = 29
then validato := validato+1;
if årtype(år) = 3 and md = 9 and dg > 2 and dg < 14
then validato := validato+1;

return validato;

end VALIDATO, Storbritannien
-----

```

Algoritme 5G: JDAG, generel-----
/* J D A G */**integer procedure** jdag(kaltype, år,md,dg);**boolean** kaltype;/* **true** for greg.kalender, **false** for jul.kalender */**integer** år,md,dg;

/* Beregner juliansk dagtal for datoen i den givne kalender, gælder ubegrænset.

jdag = 0 svarer til -4712 jan 01, juliansk eller til -4713 nov 24, gregoriansk */

begin**integer** nulldag, cyklgd, cykantal, åricyk, århunicyk;

/* beregn nulldag og cykisk længde for begge kalendere */

if kaltype = **true****then begin**

/* gregoriansk kalender */

nulldag := 1721119; cyklgd := 146097

end**else begin**

/* juliansk kalender */

nulldag := 1721117; cyklgd := 146100

end;

/* januar og februar regnes som hørende til året før */

if md < 3**then begin**

md := md + 12; år := år - 1

end;

cykantal := entier(år / 400);

/* antal cyklusser à 400 år */

åricyk := år - cykantal * 400;

/* årets placering i 400 år */

århunicyk := åricyk // 100;

/* tilhørende hundredår */

årrest := åricyk **mod** 100;

/* årets placering i 100 år */

jdag := nulldag + cykantal * cyklgd
+ (århunicyk * cyklgd) // 4 + (årrest * 1461) // 4
+ (153 * md - 457) // 5 + dg;

/* find placering fra år 0 */

/* og addér de 3 jævne */

/* interkalationer */

return jdag;**end JDAG, generel**

Algoritme 6G: DATOJ, generel

```

-----
/* D A T O J */

procedure datoj(kaltype,jdagnr,år,md,dg);
boolean kaltype; /* true for greg.kalender, false for jul.kalender */
integer jdagnr;
/* beregner dato for et juliansk dagtal i den givne kalender, gælder ubegrænset.
   jdag = 0 svarer til -4712 jan 01, juliansk, eller til -4713 nov 24, gregoriansk */

begin
integer nulldag,cyklgd,korrdag,cykantal,dagicyk,dagiårhun,århunicyk,årrest,dagrest;

/* beregn nulldag og cyklgd for den relevante kalender */

if kaltype = true
then begin /* gregoriansk kalender */
  nulldag := 1721119; cyklgd := 146097
end
else begin /* juliansk kalender */
  nulldag := 1721117; cyklgd := 146100
end;

korrdag := jdagnr - nulldag; /* relativt til år 0 feb 29 */

cykantal := entier((korrdag - 1) / cyklgd); /* antal cyklusser à 400 år */
dagicyk := korrdag - cykantal * cyklgd;

århunicyk := (4 * dagicyk - 1) // cyklgd; /* århundrede i cyklus */
dagiårhun := ((4 * dagicyk - 1) mod cyklgd) // 4 + 1; /* dagantal i 100 år */

årrest := (4 * dagiårhun - 1) // 1461; /* år i 4-års cyklus */
dagrest := ((4 * dagiårhun - 1) mod 1461) // 4 + 1; /* dagantal i 4 år */

md := (5 * dagrest + 456) // 153; /* måned, dag og år */
dg := ((5 * dagrest + 456) mod 153) // 5 + 1;
år := cykantal * 400 + århunicyk * 100 + årrest;

/* ret årstal for januar og februar */
if md > 12
then begin
  md := md - 12; år := år + 1
end;

return år,md,dg;

end DATOJ, generel
-----

```

11. Eksempler på algoritmer for andre lande

Tillæg A. Litteraturhenvisninger

1. Robert G. Tantzén: "Conversions between calendar date and julian day number". CACM algorithm no. 199, Communications of the ACM 1963, 1964.
2. Donald E. Knuth: "The calculation of easter". Communications of the ACM, 1962.
3. J. P. Søltoft: "Conversion algorithms between julian day numbers and calendar dates". Datalogisk institut, Københavns universitet, 1975. Index 75-3-1.
4. "Revised report of the algorithmic language ALGOL 60", Corrected reprint, A/S Regnecentralen, København 1964.
5. J. Fr. Schroeter: "Haandbog i kronologi", Cammermeyers boghandel, Oslo 1926. (Del I 1923, del II 1926).
6. G. V. Coyne, M. A. Hoskin and O. Pedersen: "Gregorian reform of the Calendar". Proceedings of the vatican conference to commemorate its 400th anniversary 1582-1982. Pontifica academia scientiarum, Specola vaticana 1983.
7. R. W. Bauer: "Calender for Aarene fra 601 til 2200". Genoptryk ved Dansk-historisk fællesforening, Viborg 1974.
8. David Ewing Duncan: "The calendar", Fourth Estate, London 1998, ISBN 1-85702-721-3
9. Johannes Steenstrup: "Tidsregning". En kort oversigt over kronologien til vejledning ved historisk studium, 2. udgave. H. Hagerups forlag, København 1922
10. Herluf Nielsen: "Kronologi". Dansk Historisk fællesforeningers håndbøger, København 1974, ISBN 87-7423-027-1
11. Frank Parise: "The book of calendars". Facts on file, New York 1982. ISBN 0-98196-467-8
12. "Nordisk astronomisk tidsskrift", årgangene 1916-1925.
13. "Encyclopædia Britannica", 15. udgave Chicago/London 1985, ISBN 0-85229-423-9 Macropædia Volume 15, Knowledge in depth Artiklen: Calendar, pp 460 – 477
14. "ALGOL 6, user's manual", 1st edition, 2nd printing, A/S Regnecentralen, København 1974. ISBN 87-7557-018-1
15. "OS/VS TSO Command Language Reference", VS2 Release 3.8. IBM, januar 1980, IBM no GN28-4699, GC28-0646-4
16. Elliot I. Organick: "A FORTRAN IV Primer", 2nd Printing. Addison-Wesley, August 1967.

Tillæg A. Litteraturhenvisninger

17. "Reference manual for the Ada programming language",
Proposed standard document
Cii Honeywell Bull, July 1980.
18. R. Graham, Donald E. Knuth and O. Patashnik: "Concrete mathematics",
2nd edition. Addison-Wesley, 1994. ISBN 0-201-55802-5.
19. DS/ISO-standard 8601: "Dataelementer og udvekslingsformater"
– Informationsudveksling – Angivelse af dato og tid, 2.udgave 2001-12-11.
Dansk Standard, København. Reference no ISO 8601:2000(E).
20. Paul Ahnert: "Beobachtungsobjekte für Liebhaberastronomen"
Johann Ambrosius Barth Verlag, Leipzig 1961
21. André Danjon: "Astronomie generale"
Editeurs J. & R. Sennac, Paris 1959
22. Taget ned fra Internettet 6.marts 1998 fra <http://ecuvax.cis.ecu.edu/~Pymccart/inter-grav-text.htm>
23. Taget ned fra Internettet 9.juni 2004 fra
<http://www.bluewaterarts.com/calendar/NewInterGravissimas.htm>
24. Hermann Mucke & Jean Meeus: "Canon of solar eclipses -2003 to +2526", 2nd edition
Astronomisches Büro, Wien, 1983
25. Jan Teuber: "Lige et skudsekund", artikel i
"Aktuel astronomi", årgang 2004 nr 2, ISBN 0905-8958
26. "Hvem-hvad-hvor 1981"
Politikens forlag, København 1980, ISBN 87-567-3332-1
27. "Almanak, skriv- og rejsekalender", årgangene 1980-2004
Observatoriet, Københavns universitet
28. "Dend nye forordnede Almanak for Aar efter Christi fødsel 1700"
Genoptryk fra Ole Rømer museet, Kroppedal, 2004

Af supplerende værker, der har været inspiration, skal nævnes:

Jerome T. Murray and Marilyn J. Murray: "Computers in crisis".
How to avert the coming worldwide computer systems collapse.
Petrocelli books, New York 1984. ISBN 0-89433-223-6

Donald E. Knuth: "The art of computer programming",
vol 1. 2nd printing. Addison-Wesley, 1969. ISBN 0-201-03801

G. M. Hardy and E. M. Wright: "An introduction to the theory of numbers".
5th edition. Clarendon Press, Oxford, 1979. ISBN 0-19-853171-0

Tillæg B. Liste over kronologiske begreber

Denne liste er langtfra udtømmende og skal blot give en definition og kort forklaring på de kronologiske og astronomiske begreber, som er anvendt i rapporten.

anomalistisk måned

Tidsrummet fra Månen i sin baneellipse er nærmest Jorden (perigæum) til den atter er nærmest Jorden. Middellængden er ca 27,5546 dage [21]. Den anomalistiske måned er lidt længere end både den sideriske og tropiske måned, og derfor flytter perigæet sig fremad - østover - i banen. En rundtur tager ca 8,85 år.

anomalistisk år

Tidsrummet fra Jorden i sin baneellipse er nærmest Solen (perihelium) til den atter er nærmest Solen. År 2000 er middellængden ca 365,2596 dage [20], og periheliet ligger omkring 4. januar, hvor Jorden efter Keplers love bevæger sig hurtigst. Det bevirker at forår og sommer for den nordlige halvkugle i øjeblikket er ca 7 dage længere end efterår og vinter. Da det anomalistiske år er lidt længere end både det sideriske og det tropiske år, flytter periheliet sig langsomt fremad i banen.

astronomisk månefase

Den månefase, der måles nøjagtigt astronomisk med instrumenter. Månens bevægelse om Jorden er ret uregelmæssig, da Solens masse har stor indflydelse på Måne og Jord (3-legeme problemet), og tidspunkterne for faserne vil derfor svinge en del omkring middelværdierne.

atomtid, TAI

TAI er strengt ensartet og fremadskridende uden justeringer, og regnes altid til 0°-meridianen. Tidligere blev den betegnet som ET, efemeridetid. Ved nøjagtig beregning af sol- og måneformørkelser er man nødt til at anvende TAI.

bunden månekalender

En kalender, der har et år på 12 måne-måneder, men som for at holde trit med det tropiske år af og til indsætter en 13. skudmåned i et år. Den hebraiske kalender er et eksempel på en bunden månekalender.

cyklisk månefase

Den månefase, der baserer sig på tabeller og anvender middelværdierne for tidspunkterne for månefaserne. I den nu gældende (dvs. den gregorianske) kalender kan den cykliske månefase afvige op mod 2 dage fra den astronomiske månefase.

drakonisk måned

Set fra Jorden det tidsrum hvor Månen krydser Solbanen (Ekliptika) i opadgående retning (kaldet opadstigende knude) til den atter gør det. Månebanen hælder i gennemsnit lidt over 5° mod Solbanen. Når det samtidig er tæt på nymåne eller fuldmåne, vil der indtræffe en sol- hhv. måneformørkelse. Middellængden er ca 27,2122 dage [21] og er lidt kortere end en siderisk eller tropisk måned, da knuderne drejer sig vestover og kommer rundt på 18,6 år.

efemeridetid, ET

En tidligere betegnelse for atomtid. Begrebet opstod, da man opdagede at Jorden ikke roterer strengt regelmæssigt i tid om sin akse.

ekliptika

Solbanen/Jordbanen. Set fra Jorden Solens bane rundt på himmelkuglen i løbet af et år, der afspejler Jordens bane om Solen. Ekliptika hælder i øjeblikket ca 23½° mod himmelækvatoren, og hældningen svinger langsomt mellem 24½° og 21½°. Denne vinkel er skyld i de skiftende årstider med forskellige solhøjder og daglængder igennem et år på Jorden.

Tillæg B. Liste over kronologiske begreber

epakt

Betegner for en dato Månens alder, dvs. hvor mange dage der er gået fra sidste nymåne til den pågældende dato. Værdien er mellem 0 og 29 (eller 1 og 30).

forårspunkt

Set fra Jorden skæringspunktet mellem Solbanen (Ekliptika) og himmelækvatoren, hvor Solbanen stiger op over ækvatoren. I den gregorianske kalender passerer Solen i øjeblikket punktet omkring 21. marts, forårsjævndøgnet.

Forårspunktet er udgangspunkt for koordinatsystemer for himmellegemerne.

fri månekalender

En kalender, der altid har et år på 12 måne-måneder, og som derfor er ca 11 dage kortere end et solår. Årene i en fri månekalender vil derfor flytte sig i forhold til solåret. 33 solår svarer ca til 34 år i en fri månekalender. Den islamiske kalender er et eksempel på en fri månekalender.

fuld måne-måned

En måne-måned på 30 dage i en månekalender.

gregoriansk kalender

Vores nugældende kalender med 97 skudår i 400 år. Årets gennemsnitlige længde er 365,2425 dage. Oprindeligt indført af den romersk-katolske kirke i 1582, i Danmark 1700.

gyldental

Årets nummer i den 19-årige Metoncyklus, hvorefter månefaserne omtrent falder på de samme datoer. Værdier mellem 1 og 19.

himmelækvatoren

projektion af Jordens Ækvator på himmelkuglen. Ækvator står vinkelret (90°) på Jordens rotationsakse. På grund af præcessionen forskubber himmelækvatoren sig langsomt mellem stjernerne.

hul måned

En måne-måned på 29 dage i en månekalender.

indiktion

Årets nummer i en 15-årig cyklus, indført i Romerriget til skatteudskrivninger. Værdier mellem 1 og 15.

interkalation

Indsættelse/indskud af ekstra dage eller måned for at holde en kalender i trit med solåret eller måneåret

juliansk dagnummer

En datos fortløbende dagnummer inden for en juliansk periode, startende med år -4712 jan 01, juliansk kalender som dag nr 0. Foreslået af J. J. Scaliger i 1583.

juliansk kalender

Vores tidligere kalender med 1 skudår hvert 4. år. Årets gennemsnitlige længde er 365,25 dage. Indført af Julius Cæsar i år -44 (= 45 f.v.t). Bruges nu næsten ikke mere, da året er for langt i forhold til det tropiske år.

juliansk periode

Hænger tæt sammen med juliansk dagnummer. Perioden er ligeledes foreslået af J. J. Scaliger. Længden er $28 \cdot 19 \cdot 15$ julianske år = 7980 år = 2914695 dage.

Tillæg B. Liste over kronologiske begreber

lunation

Tidsrummet fra en bestemt månefase til den næster af samme slags (fx nymåne til næste nymåne). Længden er den samme som en synodisk måned.

Metoncyklus

Den 19-årige cyklus, hvor 235 synodiske måneder, ca 6939,6882 dage, omtrent svarer til 19 julianske år, 6939,75 dage. Efter 19 år vil månefaserne stort set falde på de samme datoer i året. Benyttes både i den julianske og den gregorianske kalender til beregning af Påskedag.

Til sammenligning er 19 tropiske år ca 6939,6018 dage, og 19 gregorianske år 6939,6075 dage.

måne-måned

Er i en månekalender enten på 30 dage (fuld måne-måned) eller 29 dage (hul måne-måned).

præcession

Den langsomme drejning af Jordens rotationsakse i en tilnærmet lillecirkel omkring Ekliptikas (solbanens) poler, hvor den halve topvinkel svinger mellem ca $24\frac{1}{2}^\circ$ og $21\frac{1}{2}^\circ$ - i øjeblikket er den ca $23\frac{1}{2}^\circ$. Det er den samme vinkel som Ekliptikas hældning mod himmelækvatoren. På stjernehimlen ser det ud som om forårs-punktet langsomt glider vestover i Solbanen. Bevægelsen er ca $50,26''$ pr. år. På en menneskealder på 70 år bliver det til ca 1° (~ 2 månediameter).

Præcessionen bevirker, at stjernernes koordinater ændrer sig med tiden. Den hellenistiske astronom Hipparchos opdagede den år ca -150 ved at sammenligne sine egne observationer med ældre fra Babylonien.

påskegrænse

For et bestemt år den seneste dato efter forårsjævndøgn 21.marts, som Påskedag ikke kan falde på, dvs. dagen efter er den først mulige dato for Påskedag.

Grænsen er bestemt af den cykliske månefase*, epakt eller evt. gyldental.

Saroscyklus

Består af 223 synodiske måneder, ca 6585,3212 dage (18 år 10 dage / 18 år 11 dage), der næsten svarer til 242 drakoniske måneder, ca 6585,3567 dage og 239 anomalistiske måneder, ca 6585,5375 dage [21]. En formørkelse vil derfor næsten gentage sig efter dette tidsrum $\frac{1}{3}$ længere vestpå på Jorden. I hvert fald opdaget af babylonierne omkring år -800.

siderisk måned

Tidsrummet for Månens omløb i forhold til stjernerne. Middellængden er ca 27,3217 dage [21].

siderisk år

Tidsrummet for Jordens omløb i forhold til stjernerne. Middellængden er ca 365,2564 dage [21].

skudsekund

De sekunder, der er indsat for at holde et tilnærmet antal hele sekunder i forskel mellem atomtid, TAI og vores almindelige UT, Greenwich tid, der er uregelmæssig og følger Jordens rotation. Indsættelserne sker bagud efter målinger af afvigelserne ved måneokkultationer af stjerner.

solcirkel

Årets nummer i den 28-årige cyklus, hvorefter ugedagene gentager sig. I den julianske kalender ligger den 28-årige cyklus fast, i den gregorianske brydes den de 3 gange på 400 år, hvor der ikke er skudår.

Værdier mellem 1 og 28.

solår

Tidsrummet for et omløb af Jorden om Solen (eller set fra Jorden et omløb af Solen). Solåret kan være siderisk, tropisk eller anomalistisk.

Tillæg B. Liste over kronologiske begreber

synodisk måned

Tidsrummet for Månens omløb i forhold til Solen. Også kaldet en lunation.

Det er dette tidsrum, der bestemmer månens faser, og som alle månekalendere skal forholde sig til.

Middellængden er ca 29,5306 dage [21].

søndagsbogstav

Angiver for et år hvilken ugedagskombination det har. Bogstav A svarer til søndag 1.januar. I et skudår anvendes 2 bogstaver, et til januar-februar og et andet til marts-december, fx ED.

Oprindeligt fik hver eneste dag sig eget bogstav gennem hele året, men nu anvendes søndagsbogstavet næsten udelukkende på årsbasis.

Værdier: {A, B, C, D, E, F, G, BA, CB, DC, ED, FE, GF, AG}. Overgangsårsår kan have andre kombinationer.

tropisk måned

Tidsrummet for Månens omløb i forhold til forårspunktet. Det er på grund af præcessionen lidt kortere end en siderisk måned. Middellængden er ca 27,3216 dage [21].

tropisk år

Vores årstidsår. Tidsrummet for Jordens omløb i forhold til forårspunktet (og dermed stjernernes koordinat-system, der regnes derfra).

Det er dette år, som bestemmer vores årstider og som alle kalendere skal forholde sig til. Det er lidt kortere end det sideriske år på grund af præcessionen. Middellængden er i øjeblikket ca 365,2422 dage [21].

universaltid, UT

Vores almindelige Greenwich tid, som er uregelmæssig, da den følger Jordens rotation. Også betegnet UTC.

Tillæg C. Oversættelse af bullen: "Inter Gravissimas"

Indledende Bemærkninger

Kilderne til den latinske tekst er [22] og [23]. Den er oversat af lektor, cand.mag Christian Bohr, og forfatteren har efterfølgende redigeret og sammenholdt den med en ældre oversættelse til norsk bokmål fra [5].

Generelt er det tilstræbt at lave en lettere læselig oversættelse af den meget knudrede latinske tekst med mange indskudte sætninger. Visse sætninger er blevet brudt om eller forenklet, og nogle passiver erstattet af aktiv form. Der er gjort udstrakt brug af fodnoter til uddybende forklaringer.

Dateringer med "Kalender" er i den gammelromerske kalender, hvor der i sidste halvdel af en måned tælles ned mod den næste.

Gregors forgænger, Pius V (1566-1572), havde allerede udsendt det i bullen nævnte breviar, der indeholdt rettelser til månetabellerne.

Fortale til bullen

Godkendelse af den renoverede kalender med henblik på fejring af fester og de tilhørende gudstjenester, og afskaffelse af den gamle kalender.

Gregor, biskop⁵, Guds tjeneres tjener⁶, til evig ihukommelse:

Inter Gravissimas

Blandt de mest alvorlige pligter i Vort embede som sjælesørger er ikke mindst den, at de opgaver, som er blevet pålagt det apostoliske sæde af det hellige Tridentinske koncil (1545-1563)⁷, med Guds hjælp bliver ført til den ønskede afslutning.

1.

Ved en beslutning overlod samme koncils fædre meget rimeligt hele sagen til den romerske Pontifex's⁸ autoritet og dømmekraft, da de også på grund af breviarets øvrige spørgsmål var kommet i tidsnød.

2.

Breviaret indeholder først og fremmest to ting. Den ene omfatter de bønner og taksigelser på festdage og foregående festaftener, den anden angår påsken og de tidspunkter der afhænger af den, som skal beregnes ud fra Solens og Månens bevægelse.

3.

Vores forgænger Pius V – som Vi har i taknemmelig erindring – har sørget for at den første del blev offentliggjort og gennemført.⁹

4.

Men den anden - som naturligvis kræver en lovformelig udbedring af kalenderen – havde Vore forgængere af de Romerske Pontifices allerede forsøgt at gennemføre, men det har ikke kunnet lade sig gøre indtil nu, fordi reglerne for en kalenderreform - sådan som den var foreslået af de kyndige i himmellegemernes bevægelse – på grund af de store og næsten uovervindelige vanskeligheder, som den slags reformer altid medfører, hverken havde varig gyldighed eller bevarede de gamle kirkelige riter uændret, og det måtte i denne sag have Vores primære opmærksomhed.

⁵ Paven er også Roms biskop. Gregor 13. var pave fra 1572-1585

⁶ En standardbetegnelse (Servus Servorum Dei) for paveembedet

⁷ Det store modreformatoriske koncil (kirkemøde), der udstak den romersk-katolskes kirkes afvisende kurs overfor reformationerne

⁸ også én af pavens titler (Pontifex Maximus – ypperstepræst), overtaget fra det hedenske Rom

⁹ Med bullen "Quod a Nobis" 1568.

5.

Mens Vi altså også i denne sag har arbejdet med omtanke, omsorg og i tillid til Guds styrelse, og da sagen nu engang er blevet betroet Os, uden at Vi egentlig kan kalde os kompetente, har vores kære søn Antonius Lilius, doktor i de frie kunster og i medicin, bragt Os en bog, som hans bror Aloysius tidligere havde skrevet. I bogen har han påvist at alt det, der var kommet i uorden i kalenderen, kunne udbedres med en ny epaktcyklus*, som han selv havde udtænkt og indrettet både efter bestemt norm for selve gyldentallet* og tilpasset hvert solårs* længde. Genopretningen vil bevirke at kalenderen ikke selv senere skulle ændres yderligere.

Denne ny måde til genopretning af kalenderen har Vi sammenfattet i en lille bog, der for få år siden er blevet sendt til alle kristne¹⁰ fyrster og mere betydende universiteter, så denne sag, der er fælles for alle, også vil kunne gennemføres ved fælles rådslagning imellem os. Da de - som vi så inderligt havde ønsket det - havde svaret positivt, samlede Vi tilskyndet af den enige tilslutning sagkyndige i denne hellige by Rom. Vi havde langt tidligere udpeget dem fra den kristne¹¹ verdens førende nationer, netop med henblik på kalenderens korrektion. Efter de havde anvendt megen tid og omhu på dette studium og sammenlignet dels gamle, dels nyere cyklusser hentet alle vegne fra, og havde analyseret dem på det omhyggeligste, valgte de på grundlag af egen og andre lærde menneskers vurdering – og hvor lærde havde skrevet om denne sag - denne epaktcyklus, hvor de tilføjede ikke så få ting, som ved nøjagtig gennemsyn syntes at kunne bidrage til at gøre kalenderen fuldkommen.

6.

Da Vi altså ser at til den rette højtideligholdelse af påsken - i overensstemmelse med bestemmelserne fra kirkefædrene, de romerske biskopper, især Pius I og Victor I¹², og ikke mindst det økumeniske nikæanske koncil og andre koncilier – er det nødvendigt at sammenholde og definere tre nødvendige punkter:

For det første det nøjagtige tidspunkt for forårsjævnøgndøgn.

Dernæst datoen for den 14. månens første måned¹³, som enten falder på selve jævnøgndøgnsdagen eller kommer lige efter den.

Og endelig den til enhver tid første søndag som følger efter denne 14. måne.

Derfor har Vi ikke blot sørget for at jævnøgnet førtes tilbage til sin gamle plads, fra hvilken den var afvejet cirka 10 dage, og at den 14. påskemåne¹⁴ flyttedes tilbage til sin plads, hvorfra den nu afviger med 4 dage og endda lidt mere, men også for at videregive den måde og beregning, som sørger for at jævnøgndøgn og den 14. måne for eftertiden aldrig flyttes fra deres korrekte pladser.

7.

For at altså forårsjævnøgndøgn, som af det nikæanske koncils fædre var fastsat til den 12. Kalender april¹⁵, bliver flyttet tilbage til dette nævnte tidspunkt, så indskærper og forordner Vi at der ud af oktober måned 1582 tages i alt 10 dage fra og med 3.dagen før Nonen i oktober¹⁶ til dagen før Idus i oktober¹⁷, og at den dag som følger efter den som plejer at fejres som Skt. Franciscus' fest på den 4.dagen før Nonen¹⁸, kaldes oktobers Idus¹⁹, og at der på den dag fejres festerne for martyren Skt. Dionysius, Rusticus og Eleutherius, sammen med ihukommelsen af St.Marcus, pave og bekender, og af martyrerne Skt. Sergius, Bacchus, Marcellus, Apuleius; og at der på den 17. Kalender november²⁰, som følger nærmest efter, fejres festen for Skt. Callistus, pave og martyr; og at der på den 16. Kalender november²¹ skal ske den til den 18. søndag efter pinse hørende gudstjeneste og messe idet søndagsbogstavet* G skal ændres til C,

¹⁰ der menes i modreformatorisk ånd: alle romersk-katolske fyrster

¹¹ naturligvis kun romersk-katolske

¹² De 2 paver var i oldkirken med til at sikre, at Påskedag altid blev fejret på en søndag, jfr. kapitel 1 og 7

¹³ datoen for 14.dagen efter nymåne, dvs. fuldmåne

¹⁴ 14.dagen i påske måne-måned

¹⁵ 21.marts

¹⁶ 5.oktober

¹⁷ 15.oktober

¹⁸ 4.oktober

¹⁹ 15.oktober

²⁰ 16.oktober

²¹ 17.oktober

Tillæg C. Oversættelse af bullen: "Inter Gravissimas"

og til sidst på den 15. Kalender november²² fejres evangelisten Skt. Lucas' fest, og derefter gennemføres de øvrige festdage, som de er beskrevet i kalenderen.

8.

For at nu ingen skal komme ud for en uretfærdighed ved at disse 10 dage bliver trukket fra, skal det i de sagsanlæg, der måtte opstå som resultat heraf, være dommernes pligt at medregne den nævnte fratrækning ved at lægge 10 dage til terminen for enhver ydelse.²³

9.

For ikke at jævndøgnet igen senere skal flytte sig fra 12. Kalender april²⁴, bestemmer Vi at bissextum²⁵, som traditionen er, skal fortsætte hvert fjerde år, undtagen hvert 100. år, som har været skudår indtil nu, dog vil Vi at også år 1600 skal være skudår. Efter dette derimod skal de ikke alle være skudår, men for hvert spand af 400 år skal de tre første hundredår ikke være skudår, men den fjerde være det, dvs. at årene 1700, 1800 og 1900 ikke er skudår, men at der i år 2000 indskydes en skuddag på den vante måde og februar derfor kommer til at indeholde 29 dage, og videre at proceduren med at unnlade og indskyde skuddage i 400 års perioden bevares for steds.

10.

For endvidere at kunne finde 14. dagen²⁶ før påsken kan på rigtig måde, og ligeledes for også at Månens dage²⁷ stadig - sådan som de i tilslutning til kirkens gamle skik ved at finde dem i martyrologiet for hver enkelt dag - kan bekendtgøres korrekt for den troende menighed, beslutter Vi at gyldentallet* skal fjernes fra kalenderen og erstattes med den epaktcyklus*, der ovenfor nævnt er rettet imod gyldentallets norm. Det bevirker at nymåne og 14. dagen før påske altid beholder deres rigtige pladser. Og det fremgår tydeligt af forklaringen til vores kalender, hvor påsketavlerne er beskrevet i overensstemmelse med kirkens tidligere ritus, og derved finder man sikrere og nemmere den hellige påske.

11.

Endelig er det også nødvendigt at afbryde den for den Romerske kirke indtil i dag vante cyklus af søndagsbogstaver* indenfor 28-års perioden²⁸, dels på grund af de 10 fratrukne dage i oktober i året 1582 (som retteligen bør kaldes et rettelsens år), dels på grund af de 3 dage som indenfor tidsrummet af 400 år mindre skal indskydes²⁹. Vi vil derfor, at der i dennes sted skal indsættes den samme cyklus på 28 år, som er tilpasset af Lilius, både efter den nævnte kalkulation med at udelade en skuddag hvert hundrede år, og til den givne størrelse af solåret. Herved kan man med solcirklen* lige så let som tidligere finde søndagsbogstavet*, sådan som det udredes i selve kanonen³⁰.

12.

Altså godkender Vi ved dette dekret den ved Guds uendelige nåde overfor sin kirke korrigerede og endelige kalender i kraft af den myndighed der plejer at være Pontifex Maximus' privilegium og forpligtelse, og Vi har befaleet at den sammen med martyrologiet trykkes i Rom og efter trykning bliver offentliggjort.

13.

Men for at både martyrologiet og dekretet uforfalskede og rensat for mangler og fejl kan præsenteres overalt på Jorden, forbyder Vi alle bogtrykkere, der hører hjemme i Vort eller et under S.E.R.³¹ enten indirekte eller direkte sorterende område, under trussel om konfiskation af bøgerne og indbetaling af 100 gulddukater til

²² 18. oktober

²³ Paven var også verdslig fyrste i Kirkestaten, der eksisterede til 1870, og i 1929 blev genoprettet som Vatikanstaten

²⁴ 21. marts

²⁵ skuddagen – i den gamle kalender indsat (interkaleret) mellem 24. og 25. februar

²⁶ påskegrænsen*, dvs. 1. fuldmåne efter forårsjævndøgn – 14 dage efter nymåne, som epakten* angiver

²⁷ de cykliske månefaser*

²⁸ solcirklen*

²⁹ der menes at 3 dage skal fjernes

³⁰ den medfølgende fortegnelse

³¹ Sanctae Ecclesiae Romanae, dvs. den hellige romerske (katolske) kirke

Tillæg C. Oversættelse af bullen: "Inter Gravissimas"

den pavelige kasse; og forbyder andre hvor de end måtte befinde sig på Jorden under trussel om ekskommunikation³² og andre straffe efter vores forgodtbefindende, hvis de uden vores tilladelse vover eller bare tænker på enten at trykke og fremlægge eller at tage imod denne kalender med martyrologiet – hvadenten de er sammen eller hver for sig.³³

14.

Vi fjerner og ophæver således fuldstændig enhver gammel kalender, og Vi vil at alle patriarker, primaser, ærkebiskopper, biskopper, abbeder, og alle øvrige kirkemyndigheder introducerer den nye kalender (med den hertil passede martyrologi) ved gudstjenester og ved fejring af fester i deres respektive kirker, klostre, konventer, ordener, ridderskaber og bispedømmer, og udelukkende bruge denne, både selv og de øvrige præster og sekulære- og klostergejstlige af begge køn. Ikke at forglemme soldater og troende, som vil begynde at bruge den efter de 10 dage er fjernet fra oktober måned i året 1582. Men for dem som bebor så fjerne egne at de ikke før den foreskrevne tid kan nå at få kendskab til dette brev, skal det være tilladt i den samme oktober måned det følgende år eller endnu senere, så hurtigt nemlig som dette Vort brev måtte nå frem, på den af Os ovenfor angivne måde at foretage en sådan ændring, og således som det vil blive mere udførligt forklaret i Vort kalendarium for korrektionsåret.

15.

I overensstemmelse med den os af Gud betroede autoritet opfordrer vi og beder vores kære søn i Kristus, vores Rudolf, romernes glørværdige konge og valgt til kejser³⁴, og de øvrige konger, fyrster og republikker³⁵, og vi pålægger samme at de med den samme iver som de har ønsket af Os, at Vi gennemførte dette store arbejde, med samme, ja med endnu større iver - for at bevare fællesskabet mellem kristne³⁶ nationer under fejringen af fester - både selv antager denne Vor kalender og sørger for at den i tro antages og fuldt ud følges af alle de folk, der er underlagt dem.

16.

Men da det vil være vanskeligt at få udsendt dette brev til alle steder i den kristne³⁷ verden, forordner Vi at det offentliggøres og hænges op på apostelfyrstens kirke³⁸ og på det apostolske kancelli samt på hjørnet af Campus Floras³⁹, og Vi indskærper endvidere at man viser samme utvetydige lydighed overfor eksemplarer af dette brev, både som trykte eller hæftede foran eller inde i bind af kalendariet eller martyriologiet, og det hvadenten de er trykt eller udskrevet af den offentlige notar og forsynet med seglet fra en kirkelig autoritet.

17.

Det er derfor ikke tilladt noget menneske at skade eller forvovent at turde imødegå denne skrivelse med Vore forordninger, befalinger, statutter, vilje, approbering, fjernelse, ophævelse, opfordring og ønske. Hvis nogen alligevel påtænker at angribe dette skal han vide, at han vil pådrage sig den almægtige Guds og hans hellige apostle Peter og Pauls vrede.

Givet i Tusculum i det Herrens år 1581⁴⁰ 6. Kalender marts⁴¹, i vores pontifikats 10. år.

³² udstødelse af det kirkelige fællesskab, hvad der var meget alvorligt under modreformationen

³³ Forbudet blev senere tilbagekaldt. Paven havde givet en trykker privilegium på udgivelse, men trykkeren magtede ikke at få trykt kalender og martyriologi, så andre måtte træde til !

³⁴ Rudolf 2. af Habsburg, tysk-romersk kejser 1576-1612

³⁵ alle romersk-katolske

³⁶ romersk-katolske

³⁷ romersk-katolske

³⁸ Peterskirken i Rom

³⁹ Blomstertorvet

⁴⁰ Årstal efter florentinsk stil, hvor det nye årstal ikke begyndte 1.januar, dvs det korrekte år er 1582, jfr. bemærkningerne i kapitel 1

⁴¹ 24. februar

Supplerende deklARATION for dem, der ikke gik over i 1582

Her følger en deklARATION om overholdelse af den nyligt udgivne kalender for dem, som ikke var begyndt at bruge den fra oktober måned 1582.

Vi ønsker, i overensstemmelse med omsorgspligten i den os af Gud pålagte pastorale opgave, at den ved Vor omsorg og befaling for nylig korrigerede og udgivne kalender overalt af alle dem, som på grund af forhindringer ikke er begyndt at overholde den på den foreskrevne dag i den sidst forgangne oktober måned, hurtigst muligt bliver taget i brug og så vidt det er mest hensigtsmæssigt bliver overholdt omhyggeligt og nøjagtigt. Og Vi vil derfor - i overensstemmelse med den i Vor overvågens indeholdte forudseenhed - hurtigt tilstille hjælp mod disse vanskeligheder, for ikke at den følgende oktober måned skal afventes, hvilket trods alt vil være alt for lang tid.

1.

På eget initiativ, og efter sikker viden og moden overvejelse fra Vor side og ved den apostoliske magtfuldkommenhed, fastslår vi denne Vor for evigt gældende forordning, indskærper og forordner, at enhver som i hvilket som helst rige, provins, by eller sted endnu ikke er begyndt at overholde den omtalte, korrigerede kalender, efter måden og proceduren som er beskrevet nedenfor uden nogen undskyldninger og forhindringer følger denne kalender på denne måde:

Når man er kommet til den 10. februar i året 1583, i hvilken Quinquagesimas søndag⁴² falder efter den gamle kalender, foretager man straks en overgang til 21. i samme februar måned med udeladelse af de 10 dage, der ligger mellem den 10. og 21. dag i februar. Det vil være den anden periode⁴³ efter Quinquagesimas søndag (med søndagsbogstavet* F forandret til B). I denne anden periode skal der holdes gudstjeneste og messen for Skt. Valentin, præst og martyr, med påmindelse af Skt. Faustinus og Iuvita, martyrer, og Skt. Simeon, biskop og martyr. Og den tredje periode, om Lærerembedet. Den fjerde periode vil være Dies Cinerum⁴⁴, og på denne måde vil der ikke mere være afvigelse i fejringen af hverken de bevægelige eller de faste fester.

2.

Men da det ville være svært at sende dette brev til alle steder i den kristne⁴⁵ verden, vil Vi at der alle folk og på alle steder viser de antagne forordninger - der er trykte, underskrevne med notarius publicus's hånd, og signerede med seglet fra en person forlenet med kirkelig autoritet - samme ubetvivlelige autoritet som vises mod det originale brev, når det fremvises eller opsættes.

Givet i Rom hos S. Peter, under Piscators segl, den 10. november 1582, i vores pontifikats 11. år.

⁴² Fastelavns søndag

⁴³ med periode må her forstås dag

⁴⁴ Askeonsdag, onsdag efter Fastelavn. Dagen indleder fasten op til Påske. I 1583 23. februar efter den ny kalender

⁴⁵ her menes atter den romersk-katolske

Tillæg C. Oversættelse af bullen: "Inter Gravissimas"

Tillæg D. Forordning om gregoriansk kalender i Danmark

Kilden til forordningen er [28], der på forsiden har tilføjelsen:

”...Med hosføyede Kongel. Forordning, hvor ved Alle De for samme Aar allerede tryckte Almanacker gandske afskaffis.”

Ole Rømers argumenter overfor kongen for overgangen er angivet i [6] i engelsk oversættelse.

Vi Friderich dend Fierde,

af Guds Naade Konge til Danmarck og Norge, de Venders og Gothers, Hertug udi Slesvig, Holsten, Stormarn og Ditmarsken, Greve udi Olden- borg og Delmenhorst

Gjøre alle vitterligt:

at Vi allernaadigst haver for got befunden at paabyde og anordne, saasom Vi og hermed paabyde og anordne, at udi indkommende Aar 1700 Februarii Maanet ej skal have uden 18 Dage, saa at nest efter den 18 Februarii skal tællis den 1 Martii, og saaledis Paaske Dag i samme Aar indfalde den 11 April, hvor efter de rette Nye Almanacker skal indrettis; Og maa ingen andre Almanacker⁴⁶ brugis udi Vore Riger og Lande end samme Ny Almanacker, hvorefter alle Breve og Documenter efter denne Dag skal dateris og forstaas, under 4 Rdl. Straf i Kiøbstæderne og 1 Rdl. Straf paa Landet, til Stæders Fattige for hver gang nogen sig herimod forseer;

Dog hvis Vexselbreve eller andre Forskrivelser, som forhen for denne Forordnings publication kunne være udgangen, at efterkommis til en vis Dag inden nestkommende Junii Maanets udgang, maa forstaaes efter den forhen i brugte gammel Stiil, men siden skal alting efterkommis efter denne nu forordnede Ny Stiil, og derpaa indrettede Almanacker;

Skulle nogen understaa sig, denne Almanackens forandring til nogen forsætlig Svig og Bedragerie i Brevis og Documenters datering at misbruge, da skal den skyldige derfor efter Sagens befunden Beskaffenhed i høyeste maader ansees of straffis:

Hvor efter alle og enhver sig allerunderdanigst haver at rette og for Skade at tage vare;

Og byde Vi hermed og befale Voris Grever og Friherrer, Stiftsbefalingsmænd, Amtmænd, Præsidenter Borgemester og Raad, samt Byfogder og alle andre vedkommende, som denne Voris Forordning under Voris Cancelli Seigl tilskikket vorder; At de den paa behørige Stæder til allis efterretning, strax lader læse og forkynde.

Givet paa Vort Slot Kiøbenhafn, dend 28. Novembr. 1699

Under Vor Kongl. Haand og Zignet
Friderich R.

⁴⁶ Forordningen kom med datidens langsomme kommunikation i sidste øjeblik, men Frederik 4. var dog først blevet konge 25. august 1699. Der var forlængst blevet trykt almanakker for 1700 med den gamle kalender; disse gamle almanakker skulle tilintetgøres!

Tillæg D. Forordning om gregoriansk kalender i Danmark

Tillæg E. Overgang fra juliansk til gregoriansk kalender

Som nævnt i kapitel 1 skete overgangen ikke i ét hug. Når man gennemgår kilderne, er det til at rive sig i håret af fortvivlelse dels over de mange tidspunkter, og dels over at kilderne på mange punkter er modstridende eller mangelfulde. Ikke nok med det - man skal også registrere brug af egne kalendere og skift tilbage fra den gregorianske til den julianske kalender på grund af territoriale ændringer.

Enkelte kilder er dog blevet slået sammen, nemlig hvor én kilde blot angiver et årstal, mens en anden angiver præcise datoer for det år.

Alle overgange, som **ikke** er fra den julianske til den gregorianske kalender, er markeret med fede typer, og der er tilføjet supplerende bemærkninger om hvilke overgange, det i stedet for så er.

For nogle landområder er kilderne specielt problematiske:

Alsace og Strasbourg

Ved den Westfalske fred i 1648 annekterede Frankrig en del af Alsace, og i 1682 besatte Louis 14.s tropper under Reunionerne (overfald på nabostaterne) simpelthen resten sammen med den befæstede Strasbourg by.

Belgien

Var i 1582 de Spanske Nederlande. Frankrig erobrede ved krige i 1600-tallet de sydlige dele, bl.a. Artois.

Finland

Brugen af egen kalender og overgangen til den gregorianske kalender under svensk herredømme er velbelyst, men der savnes oplysninger om de 2 kalendere blev anvendt side om side i den russiske periode 1809-1917 på samme måde som i Polen. Finland erklærede sig selvstændigt i dec 1917, og den nævnte overgang i 1917-1918 er i hvert fald definitiv.

De baltiske stater

Spørgsmålet er hvor meget den specielle svenske kalender slog igennem i Estland og Livland 1700-1712. Under den store nordiske krig besatte Rusland efterhånden de 2 områder 1700-1710, men annekterede dem først officielt ved freden 1721 i Nystad (Uusikaupunki).

Kurland tilhørte Polen-Litauen til den 3. polske deling 1795, og var under Rusland til 1914. Under 1.verdenskrig flyttede fronten mellem det kejserlige Tyskland og Rusland sig langsomt østpå, og ved den (midlertidige) fred i Brest-Litovsk i marts 1918 annekterede Tyskland områderne. Overgang fra den julianske kalender til den gregorianske virker kaotisk – nærmest som i hælene på den kejserlige tyske hær.

Polen

Oplysningerne om fuldstændig overgang i 1582 er ikke pålidelige. Det har ganske enkelt ikke kunnet nås i det dengang meget store og løst organiserede rige. Først omkring 1587 ser det ud til, at den gregorianske kalender er i brug overalt i Polen. Ved Polens delinger i 1772, 1793 og 1795 og atter ved Wienerkongressen i 1815 bibeholdtes den gregorianske kalender i de preussiske og østrigske dele, men for de russiske dele nævnes det i [5] at de 2 kalendere blev anvendt side om side indtil 1.verdenskrig. Ligesom for de baltiske stater ser det ud til, at den gregorianske kalender blev enerådende i hælene på de tyske og østrigske hære mellem 1914 og 1918. Fra Polens selvstændighed i 1919 har den gregorianske kalender været gældende.

Balkan

På Balkan hersker der den største forvirring om tidspunkter for overgangen.

Den græsk-ortodokse kirke anvender stadig internt den julianske kalender, men i det daglige benyttes ”den forbedrede kalender”, som er magen til den gregorianske indtil 2800.

Efterhånden som 1.verdenskrig bredte sig på halvøen – Serbien, Montenegro og Tyrkiet fra 1914, Bulgarien fra 1915, Rumænien fra 1916, Albanien og Grækenland fra 1915/1917 – tyder meget på, at den gregorianske kalender successivt er indført under 1.verdenskrig under opholdet af alle de fremmede tropper fra vest- og centraleuropæiske lande, Australien og New Zealand.

Efter 1.verdenskrig holdt staterne officielt en konference i 1923 [8], og vedtog under en kaotisk diskussion officielt den forbedrede kalender, mens den ortodokse kirke delvis fastholder den gamle kalender.

Tillæg E. Overgang fra juliansk til gregoriansk kalender

Tabel E1. Overgang fra juliansk til gregoriansk kalender				
Landområde	sidste dato i gl. kalender	efterfulgt af første dato i ny kalender	Bemærkninger	Kilder
Albanien	1923	1923		[10]
Alsace (delvis)	1648	1648	ved fransk annekktion	[10]
	1682 feb 05	1682 feb 16	ved fransk erobring	[11]
Spanske Nederlande	1582 dec 21	1583 jan 01	i dag Belgien	[6,8,11]
	1584	1584		[8]
Spanske Nederlande, Artois	1582 dec 14	1582 dec 25	i dag del af Frankrig	[5-9,11]
Spanske Nederlande, Brabant	1582 dec 14	1582 dec 25	i dag del af Belgien	[5-9,11]
	1582 dec 21	1583 jan 01		[5-8,10-11]
Spanske Nederlande, Flandern	1582 dec 14	1582 dec 25	i dag del af Belgien	[5-9,11]
	1582 dec 21	1583 jan 01		[5-8,10-11]
Spanske Nederlande, Hainault	1582 dec 14	1582 dec 25	i dag delvis fransk	[5-9,11]
	1582 dec 21	1583 jan 01		[5-8,10-11]
Bulgarien (delvis)	1912	1912		[8]
	1915	1915		[8]
	1915 okt 31	1915 nov 13		[11]
	1916 mar	1916 mar		[8]
	1920 sep 03	1920 sep 17		[11]
Bulgarien	1923	1923		[10]
Canada	1582 dec 09	1582 dec 20	fransk koloni	[11]
	1752 sep 02	1752 sep 14	britisk koloni	[11]
Danmark	1700 feb 18	1700 mar 01		[5-11]
Estland	1915	1915	under 1.verdenskrig	[8]
	1918 feb 01	1918 feb 15		[11]
Finland	1700 feb 28	1700 mar 01	juliansk -> egen kal.	[5,6]
	1712 feb 30	1712 mar 01	egen kal. ->juliansk	[5,6]
	1753 feb 17	1753 mar 01		[5,6]
	1809	1809	greg. -> juliansk	formodet
	1917 dec 31	1918 jan 14		[11]
Frankrig	1582 dec 09	1582 dec 20		[5-11]
	1793 okt 04	II vendém.14	greg. -> revol.kal.	[5-9,11]
	XIV nivôse 10	1806 jan 01	revol.kal -> greg.	[5-7,9,11]
Færøerne	1700 nov 16	1700 nov 28		[5]
Grækenland (delvis)	1916 jul 14	1916 jul 28		[11]
	1920 mar 04	1920 mar 18		[11]
Grækenland	1923 feb 15	1923 mar 01		[5-6,10]
	1924	1924		[8]
Irland	1752 sep 02	1752 sep 14		[11]
	1782	1782		[5,10]
Island	1700 nov 16	1700 nov 28		[5]
Italien, kat.stater	1582 okt 04	1582 okt 15		[5,11]
Italien, Firenze	1749 dec 20	1750 jan 01		[5,10]
Italien, Pisa	1749 dec 20	1750 jan 01		[5,10]
Japan	egen kalender	1873 jan 01		[6,8]
	egen kalender	1893 jan 01		[11]
Kina	egen kalender	1912 feb 12		[6,8,11]
Kroatien	1923 sep 30	1923 okt 14		[5]
.... fortsættes næste side				

Tillæg E. Overgang fra juliansk til gregoriansk kalender

Tabel E1, fortsat				
Overgang fra juliansk til gregoriansk kalender				
Landområde	sidste dato i gl. kalender	efterfulgt af første dato i ny kalender	Bemærkninger	Kilder
Letland, Kurland	1582 okt 04	1582 okt 15		[6]
	1617	1617		[10]
	1796 feb 07	1796 jan 28	greg. - > juliansk	[5,10]
	1915	1915	under 1.verdenskrig	[8]
	1918 feb 01	1918 feb 15		[11]
Letland, Livland	1915	1915	under 1.verdenskrig	[8]
	1918 feb 01	1918 feb 15		[11]
Litauen	1582 okt 04	1582 okt 15		[6]
	1800 jan 11	1800 jan 01	greg. - > juliansk	[5]
	1915	1915	under 1.verdenskrig	[8]
	1918 feb 01	1918 feb 15		[11]
Lorraine	1582 dec 09	1582 dec 20	som del af Frankrig	[5,10]
	1760 feb 16	1760 feb 28	ved fransk arv	[11]
Montenegro	1919 mar 04	1919 mar 18		[8,11]
	1923	1923		[10]
Nederland, Friesland	1700 nov 30	1700 dec 12		[7,9]
	1700 dec 31	1701 jan 12		[5-9,11]
Nederland, Geldern	1700 jun 19	1700 jul 01		[9,11]
	1700 jun 30	1700 jul 12		[5,9-10]
	1700 nov 30	1700 dec 12		[7,9]
Nederland, Groningen	1700 nov 30	1700 dec 12		[7,9]
	1700 dec 31	1701 jan 12		[5-9,11]
Nederland, Holland	1582 dec 14	1582 dec 25		[5-9,11]
Nederland, Holland (delvis)	1582 dec 21	1583 jan 01		[5-8,10-11]
		først i 1700		[5]
Nederland, Overijsel	1700 jun 12	1700 jul 01		[9,11]
	1700 nov 30	1700 dec 12		[7,9-10]
Nederland, Utrecht (delvis)	1700 jun 19	1700 jul 01		[9,11]
Nederland, Utrecht	1700 nov 30	1700 dec 12		[5,9-10]
Nederland, Zeuthen	1700 jun 30	1700 jul 12		[9-10]
	1700 nov 30	1700 dec 12		[7,9]
Norge	1700 feb 18	1700 mar 01		[5-11]
Polen, pr. og østr. dele	1582 okt 04	1582 okt 15		[6,9-11]
Polen, russ. dele	1582 okt 04	1582 okt 15		[6,9-11]
	1793/1795	1793/1795	greg. - > juliansk	formodet
	1918 jan 31	1918 feb 14		[11]
Polen, Danzig	1582 okt 04	1582 okt 15		[10]
Portugal	1582 okt 04	1582 okt 15		[6-11]
Preussen, dvs Østpreussen	1610 aug 22	1610 sep 02	uden for Tyske Rige	[10]
Rumænien (delvis)	1919 mar 04	1919 mar 18		[11]
Rumænien (delvis)	1919 mar 31	1919 apr 14		[5,8,10-11]
Rusland, vest	1917 dec 31	1918 jan 14	under borgerkrigen	[11]
	1918 jan 31	1918 feb 14		[5-6,8,10,13]
Rusland, øst	1920 mar 04	1920 mar 18	under borgerkrigen	[11]
Savoyen	1582 okt 04	1582 okt 15		[8]
Schweiz, Appenzell (delvis)	1597 jan 06	1597 jan 17		[11]
Schweiz, Appenzell (delvis)	1723 dec 19	1724 jan 01		[10-11]
Schweiz, Basel	1700 dec 31	1701 jan 12		[5-7,9-11]
.... fortsættes næste side				

Tillæg E. Overgang fra juliansk til gregoriansk kalender

Tabel E1, fortsat				
Overgang fra juliansk til gregoriansk kalender				
Landområde	sidste dato i gl. kalender	efterfulgt af første dato i ny kalender	Bemærkninger	Kilder
Schweiz, Bern	1700 dec 31	1701 jan 12		[5-7,9-11]
Schweiz, Biel	1700 dec 31	1701 jan 12		[5,9,11]
Schweiz, Fribourg	1584 jan 11	1584 jan 22		[10-11]
Schweiz, Genève	1700 dec 31	1701 jan 12		[5-7,9-11]
Schweiz, Glarus	1723 dec 19	1724 jan 01		[10-11]
Schweiz, Graubünden (delvis)	1760	1760		[10]
	1812 feb 17	1812 mar 01		[10-11]
Schweiz, Luzern	1584 jan 11	1584 jan 22		[10-11]
Schweiz, Neuchatel	1700 dec 31	1701 jan 12		[5,7,9,11]
Schweiz, Schaffhausen	1700 dec 31	1701 jan 12		[5-7,9-11]
Schweiz, Schwyz	1584 jan 11	1584 jan 22		[10-11]
Schweiz, Solothurn	1584 jan 11	1584 jan 22		[10-11]
Schweiz, St. Gallen	1723 dec 19	1724 jan 01		[7,11]
Schweiz, St. Gallen, by	1724	1724		[10]
	1700 dec 31	1701 jan 12		[5,9-11]
Schweiz, Thurgau	1700 dec 31	1701 jan 12		[5,9-11]
Schweiz, Unterwalden	1584 jan 11	1584 jan 22		[10-11]
Schweiz, Uri	1584 jan 11	1584 jan 22		[10-11]
Schweiz, Valais	1655 feb 28	1655 mar 11		[10]
Schweiz, Valais (delvis)	1621 dec 21	1622 jan 01		[11]
	1656 feb 18	1656 mar 01		[11]
Schweiz, Zug	1584 jan 11	1584 jan 22		[10-11]
Schweiz, Zürich	1700 dec 31	1701 jan 12		[5-7,9-11]
Schweiz, Aargau	1700 dec 31	1701 jan 12		[5,9,11]
Serbien	1919 mar 04	1919 mar 18		[8,11]
	1923 sep 30	1923 okt 14		[5,10]
Siebenbürgen (Transsylvanien)	1590 dec 13	1590 dec 24		[5]
	1590 dec 14	1590 dec 25		[10]
Slesvig	1700 feb 18	1700 mar 01		[5]
Spanien	1582 okt 04	1582 okt 15		[5-11]
Spanske kolonier, Peru	1584 okt	1584 okt		[6]
Storbritanien	1752 sep 02	1752 sep 14		[5-11]
Strasbourg, bispedømme	1582 nov	1582 nov		[6]
Strasbourg by	1583 okt 05	1583 okt 16		[5]
	1682 feb 05	1682 feb 16	ved fransk erobring	[10]
	1682 feb 18	1682 mar 01	ved fransk erobring	[7,11]
Sverige	1700 feb 28	1700 mar 01	juliansk -> egen kal.	[5-11]
	1712 feb 30	1712 mar 01	egen kal. ->juliansk	[5-11]
	1753 feb 17	1753 mar 01		[5-11]
Tyrkiet, europæisk del	1908	1908		[11]
Tyrkiet	1914	1914		[5,10]
Tyrkiet, asiatisk del	1914	1914		[11]
Tyske rige, Augsburg	1583 feb	1583 feb		[6]
	1699 sep 22	1699 okt 03		[11]
	1700 feb 18	1700 mar 01		[5-10]
Tyske rige, Bayern	1583 okt	1583 okt		[6,8]
	1584 okt 05	1584 okt 16		[11]
.... fortsættes næste side				

Tillæg E. Overgang fra juliansk til gregoriansk kalender

Tabel E1, fortsat				
Overgang fra juliansk til gregoriansk kalender				
Landområde	sidste dato i gl. kalender	efterfulgt af første dato i ny kalender	Bemærkninger	Kilder
Tyske rige, Brandenburg	1699 nov 04	1699 nov 15		[11]
	1700 feb 18	1700 mar 01		[5-10]
Tyske rige, Hessen	1699 nov 04	1699 nov 15		[11]
	1700 feb 18	1700 mar 01		[5-10]
Tyske rige, Hildesheim	1631 mar 15	1631 mar 26		[10]
Tyske rige, Holsten	1700 feb 18	1700 mar 01		[5,6]
Tyske rige, Lausitz	1584 jan 12	1584 jan 23		[10]
Tyske rige, Mainz	1583 nov 03	1583 nov 14		[5-6,8-9,11]
	1583 okt 05	1583 okt 16		[5]
Tyske rige, Minden	1630	1630		[10]
Tyske rige, Münster	1583 nov 03	1583 nov 14		[5-6,8-9,11]
Tyske rige, Nürnberg	1699 sep 22	1699 okt 03		[11]
	1700 feb 18	1700 mar 01		[5-10]
Tyske rige, Osnabrück	1624	1624		[10]
Tyske rige, Paderborn	1585 jun 16	1585 jun 27		[10]
Tyske rige, Pfalz-Neuburg	1615 dec 13	1615 dec 24		[10]
Tyske rige, Rheinland-Pfalz	1699 sep 22	1699 okt 03		[11]
	1700 feb 18	1700 mar 01		[5-10]
Tyske rige, Sachsen	1699 nov 04	1699 nov 15		[11]
	1700 feb 18	1700 mar 01		[5-10]
Tyske rige, Schlesien	1584 jan 12	1584 jan 23		[10]
Tyske rige, svenske dele	1700 feb 18	1700 mar 01		[10]
Tyske rige, Ulm	1699 sep 22	1699 okt 03		[11]
	1700 feb 18	1700 mar 01		[5-10]
Tyske rige, Utrecht (delvis)	1700 dec 01	1700 dec 12		[11]
Tyske rige, Westfalen	1584 jul 01	1584 jul 12		[10]
Tyske rige, Würzburg	1583 okt 05	1583 okt 16		[5-6,8]
Tyske rige, Østrig (delvis)	1583 dec	1583 dec		[6]
	1583 okt 04	1583 okt 15		[5-6,8]
	1584 jan 05	1584 jan 16		[5]
	1584 jan 06	1584 jan 17		[10]
Tyske rige, Østrig, Böhmen	1584 jan 06	1584 jan 17		[9-11]
	1584 jan 07	1584 jan 18		[6]
Tyske rige, Østrig, Kärnten	1583 dec 14	1583 dec 25		[11]
Tyske rige, Østrig, Mähren	1584 okt 04	1584 okt 15		[6]
Tyske rige, Østrig, Tyrol	1583 okt 05	1583 okt 16		[11]
Tyske rige, Østrig, Steiermark	1583 dec 14	1583 dec 25		[11]
Ungarn	1582 okt 21	1582 nov 01	uden for Tyske Rige	[11]
	1584 jan 22	1584 feb 02		[10]
	1587 okt 20	1587 okt 31		[5,8-9]
USA, Alaska (delvis)	1867 okt 04	1867 okt 18	med skift over Datolinien	[6,11]
	1900 aug 06	1900 aug 20		[11]

Tillæg E. Overgang fra juliansk til gregoriansk kalender

Teknisk rapport om dato- og tidsberegninger,

Rettelsesliste november 2004 ► februar 2005

Fejl og mangler påpeget af Nils Andersen er markeret med [NA]

Forside

side 1: Rettet ”november 2004” ► ”februar 2005”
side 2: Nyt bibliotek ”kalrap200502.doc”
Tilføjet ”revideret februar 2005”

Indholdsfortegnelse

side 3-4: kapitel 3, nu kun ét underafsnit ”Heltalsregning” [NA]
kapitel 4, Tilføjet ”Egenskaber ved kalenderberegninger” [NA]
Rettet sidenumre fra kapitel 3 og ud

Sammendrag

side 5: Rettet ”heltalsdividion” ► ”heltalsdivision” [NA]

Forord

side 7: Rettet ”Rapporten behandler ... om” ► ”Rapporten handler ... om”
Tilføjet en tak til Nils Andersen for gennemlæsning og kommentarer.
Tilføjet ”revideret februar 2005”

Kapitel 1

side 9: Rettet ”omstridt, bl.a. på grund af fordi” ► ”omstridt, bl.a. fordi” [NA]
Rettet ”har netop lært den at kende den i” ► ”har netop lært den at kende i” [NA]
side 13: Rettet ”først også efter at være afskaffet” ► ”først efter at være afskaffet” [NA]

Kapitel 2

side 15: Rettet ”i steder for” ► ”i stedet for” [NA]
Fremhævet ”kunne undersøge, om en forelagt dato”

Kapitel 3

ny side 19-22: Erstatte gl side 19-28. Tabel 3.1 er omarbejdet, tabel 3.2 er udgået. [NA]

Nye sider 23-104 erstatter de gamle sider 29-110.

Kapitel 4

ny side 23-32: Erstatte gl side 29-38. [NA]
Nyt eksempel 4.5, jævn interkalation, er tilføjet [NA]
Gl eksempel 4.5 er udgået.
ny side 24: Fremhævet udtrykkene ” $(A * p) // Q$ ”, ” $A // Q$ ” og ” $A // Q + 1$ ”,
som er kernen i ideen om jævn interkalation. [NA]
(I den fremsendte rettelse er der fejlagtigt skrevet ”/” i stedet for ”//”)
Rettet ”så der kommer ... frem, så udtrykket kan reduceres”
► ”så der kommer ... frem, og derved kan udtrykket reduceres”
ny side 31 (gl 37): ”med brug af ...” er i begge algoritmer flyttet op i overskrifterne.
Fremhævet operatoren ”**mod**” i algoritme 4.

Kapitel 5

ny side 33 (gl 39): Rettet ”Kronologi kan nok bedst...læren om tidsregning”
► ”(læren om tidsregning)”
ny side 36 (gl 42): Rettet ”Århunderet” ► ”Århundredet” [NA]
Rettet ”historiske årstal skal man for astr.år < 1 ... hist.år = 1 – astr.år.”
► ”historiske årstal, skal man for astronomisk.år < 1 ... historisk år f.v.t
= (1 – astronomisk år).” [NA]
ny side 37 (gl 43): Fremhævet operatoren ”**mod**” i algoritme 5.
ny side 38 (gl 44): Fremhævet operatoren ”**mod**” 3 steder i algoritme 6.
ny side 40 (gl 46): ”med brug af ...” er i begge algoritmer flyttet op i overskrifterne.

Teknisk rapport om dato- og tidsberegninger, rettelsesliste

Kapitel 6

ny side 41 (gl 47): Udgået ”Det kan umiddelbart se meget ubehageligt ud, da der så muligvis ikke er nogen ende på, hvor mange år man skal søge igennem for at få lagt ugenumrene fast. Så galt går det nu heldigvis ikke.” [NA]

ny side 42 (gl 48): Tilføjet en mangel i tabel 6.1, sidste række 2 gange ugenr ”1” efter ugenr 53.

ny side 43 (gl 49): Erstatning. Rettede og præciserede formuleringer efter overvejelser om, hvor mange ugebasisdatoer, man er nødt til at kende. Fremhævelse af ugebasisdatoen som nul dagen i nulugen. [NA]

ny side 45 (gl 51): Erstatning. Ny algoritme 9, hvor der højst beregnes 2 ugebasisdatoer. [NA]

ny side 46 (gl 52): Rettet ”Her kan man dog nøjes med” ► ”Her skal man”.

Kapitel 7

Ingen rettelser.

Kapitel 8

Ingen rettelser.

Kapitel 9

ny side 67 (gl 73): Rettelser. Variablene ”flgår” ► ”næstår” og ”flgmd” ► ”næstm” af hensyn til læseligheden

Kapitel 10

ny side 69 (gl 75): Rettet ”... døgn, så man kan arbejde i” ► ”... døgn, så man også kan operere med”
Tilføjet 2. afsnit på siden ”Krav til nøjagtighed af beregningerne...”
som introduktion til atomtid og sommertid.

Rettet ”TAI er ... ensartet og fremadskridende...” ► ”TAI er ... ensartet...”

ny side 70 (gl 76): Rettet ”klokkeslet” ► ”klokkeslæt” 3 steder. [NA]

Rettet ”fritstillet” ► ”frit stillet” [NA]

Kapitel 11

ny side 75 (gl 81): Rettet ”melle” ► ”mellem”

Tillæg A

ny side 86 (gl 92): Litteraturhenvisning 18 er rettet til ”Concrete mathematics”. [NA]
Knuth: ”The art...” er flyttet ned under supplerende værker.

Tillæg B

ny side 90 (gl 96): Tilføjet under søndagsbogstav ”Overgangsår kan have andre værdier.”

Tillæg C

ny side 92 (gl 98): Rettet i pkt 7 ”3.dagen for Nonen” ► ”3.dagen før Nonen”

ny side 93 (gl 99): Rettet i pkt 9 ”flytte sig fra fra” ► ”flytte sig fra” [NA]

Tillæg D

Ingen rettelser.

Tillæg E

Ingen rettelser.

Rettelsesliste februar 2005 ► marts 2005

Forside

side 1: Rettet ”februar 2005” ► ”marts 2005”

side 2: Nyt bibliotek ”kalrap200503.doc”

Tilføjet ”revideret marts 2005”

Sammendrag

side 5: Rettet ”Der er angivet formler til omskrivning, hvis ...”

► ”Det påvises, at så længe man holder sig til ikke-negative værdier...”

Forord

side 7: Tilføjet ”...på Datalogisk Institut ved Københavns Universitet...”

Tilføjet ”revideret 21.marts 2005”

Kapitel 11

side 75-84: Rettet sideoverskrift ”nandre lande” ► ”andre lande”