Kuriai metų savaitei priklauso diena

Šį uždavinį galima spręsti įvairiais būdais. Kadangi jis nesunkus, duomenų nėra daug, tuomet reikia pasirinkti aiškesnį ir trumpesnį algoritmą.

Svarbiausia, ką reikia rasti šiam uždaviniui: 1) sužinoti, kelinta metų diena yra duotoji diena; 2) nustatyti pirmosios duotųjų metų savaitės trukmę. Po to apskaičiuoti savaitės numerį nesunku: tereikės pasinaudoti formule

(*d_sk* - *trukmė* - 1) **div** 7 + 2;

čia *d_sk* reiškia dienų skaičių nuo duotos datos metų pradžios iki pateiktos dienos, kitaip sakant, kelinta metų diena yra duotoji diena, o *trukmė* – pirmosios metų savaitės ilgis. Šį skirtumą sumažiname vienetu dėl to, kad septintoji savaitės diena (sekmadienis) priklauso tai pačiai savaitei (pagal lietuvišką kalendorių). Pavyzdžiui, jei sausio 7 d. yra sekmadienis, o ieškome kuriai savaitei priklauso sausio 14 diena, tai

(14 - 7 - 1) **div** 7 + 2 = 2

ir gauname, kad ieškoma diena yra antroje metų savaitėje.

Dvejeto prisumavimas reikalingas tam, kad papildytume dviem savaitėmis: pirmąja metų savaite ir paskutine savaite, kurioje yra ieškoma data. Mat operacijos **div** rezultatas yra pilnų savaičių skaičius iki savaitės, kurioje yra duota diena (t .y. paskutinė savaitė neįtraukiama).

Nustatysime duotosios dienos numerį nuo metų pradžios. Tai įprastas datų skaičiavimo algoritmas. Laikome, kad data aprašyta įrašu. Nors neegzistuoja nulinė diena ar mėnuo, tačiau nulis įtraukiamas į atkarpos tipą, aprašantį mėnesį ir dieną. Mat atliekant skaičiavimus su datomis, kartais prireikia ir ribinės reikšmės – nulio.

```
type metai = 1583..5000;   { Grigaliaus kalendorius }
     data = record
              mt: metai;
              mėn: 0..12;
              diena: 0..31
            end;
```

```
function d_sk (dt: data): integer;
{ randa duotos dienos numerį nuo metų pradžios }
  var dienos: integer;
      t: data;
  function d_mėn (dt: data): integer;
  { kiek dienų turi duotų metų duotas mėnuo }
  begin
    case dt.mėn of
      4, 6, 9, 11: d_mėn := 30;
      2: if kel (dt.mt) { jei metai keliamieji }
            then d_mėn := 29
            else d_mėn := 28;
      1, 3, 5, 7,
      8, 10, 12: d_mėn := 31
    end
  end;
begin
  dienos := 0;
  t := dt;
  for t.mėn := 1 to dt.mėn - 1 do
      dienos := dienos + d_mėn(t);
  d_sk := dienos + dt.diena
end;
```

Toliau ieškosime pirmosios metų savaitės trukmės. Pasinaudosime 2-ojo uždavinio funkcija *naujieji_metai*. Ji randa kurią savaitės dieną prasideda duotieji metai. Funkciją teks šiek tiek pakeisti (pritaikyti, kai data vaizduojama įrašu).

Pateikiame likusį algoritmo tekstą:

```
function savaitės_numeris (dt: data): integer;
{ nustato kelintai metų savaitei priklauso duotoji data }
  var trukmė: integer;
  function kel (m: metai): boolean;
    { ar metai m keliamieji }
  begin
    kel := (m mod 400 = 0) or
           (m mod 100 <> 0) and
           (m mod 4 = 0)
  end;
```

```
  function naujieji_metai (m: metai): integer;
  { randa pirmąją metų savaitės dieną }
  { savaitės dienos numeruojamos: }
 {1 – pirmadienis, 2 – antradienis, ..., 7 – sekmadienis }
    var mt: metai;  { pasirenkami metai, kurių pirmoji diena sekmadienis }
        d, i: integer;
  begin
    mt := 1978;  { pasirenkami metai }
    { kurių pirmoji savaitės diena yra sekmadienis}
    d := 7;
    for i := mt-1 downto m do { jei duoti metai yra ankstesni }
        if kel (i)
           then d := ((d - 2) + 7) mod 7
           else d := ((d - 1) + 7) mod 7;
    for i := mt to m-1 do   { jei duoti metai yra vėlesni }
        if kel (i)
           then d := (d + 2) mod 7
           else d := (d + 1) mod 7;
    if d = 0
       then d := 7;
    naujieji_metai := d
  end;

begin
  trukmė := 7 - naujieji_metai (dt.mt) + 1;
  savaitės_numeris := (d_sk(dt) - trukmė - 1) div 7 + 2
end;
```