ÖRÖKNAPTÁR

Varga Gyula

Az öröknaptár probléma gyakran felmerül az elektronikus számítógépek bevezetése folyamán a gyakorlati élet különböző területein. Gyakori tény pl., hogy a számítógép által használt dátum (monitor-date, vagy todays-date) 5 karakteres alakú, míg a gép által készített kiírási termékeken, bizonylatokon (pl. bérjegyzéken, számlákon, zárlati naplókon stb.) 6 karakteres alakra van szükség, s ennek automatikus előállítása nem lényegtelen a feldolgozás flexibilitásánál. A pénzügyi tevékenységben jelentős feladat a "lejárati dátumok" ellenőrzése, mind az esedékesség, mind a kamatok tekintetében. Vagy gondoljunk csak arra, hogy a számítógépek lehetővé teszik a termelés-programozás során a megrendelések magas színvonalon intézett, automatikus visszaigazolását. A kieső vasárnapok összeszámolásában az öröknaptár-elv segíthet. A történészek, régészek munkájában sok segítséget adhat egy össszeállított öröknaptár. Nem véletlen, hogy az irodalomban is találkozunk öröknaptár programokról szóló beszámolókkal. De publikáció nélkül is számtalan helyen készült demonstrációs célokra öröknaptár program.

A megoldás alapját szolgáló naptárelv az elmúlt évszázadokban neves matematikusokat és csillagászokat foglalkoztatott. A naptár: kísérlet a trópikus évnek* közép-napokban** való kifejezhetőségére. Elsősorban a csillagászati ismeretek bővülésével ez a kísérlet egyre eredményesebb lett. A két legismertebb naptár, a Julianus és Gergely féle. A Julianus naptár, amelyet Julius Ceasar vezetett be Sosigenes, alexandriai csillagász tanácsára i.e. 45-ben, i.u. 325-ben 3 napos módosítással 1582. október 15-ig volt érvényben. A Julianus naptár szerint egy év 365 napból áll, kivéve minden negyediket, mely 366 napos. A Gergely naptár – melyet elsősorban Lilius és Clavius alkotott meg – ezt úgy javította, hogy figyelembe véve a felgyülemlett eltolódásokat, 10 napot átugrott és az évszázadok közül csak a 4-el oszthatók maradtak szökőévek.

A továbbiakban ismertetünk egy olyan öröknaptár programot, mely időszámításunk kezdetétől határozza meg egy dátumhoz a napot.

*Trópikus év = az az időköz, mely az évi látszó mozgást végző Napnak a közepes tavaszponton való két egymásutáni átmenete között telik el.
Közép-napban való értéke = 365 nap, 5 óra, 48 p., 46 mp.

**Közép-nap = a Nap két egymást követő alsó delelése közt eltelt idő a Nap – nap. Ez nem egyenletes, a Nap változó sebességű mozgása miatt. A Nap nem egyenletes mozgásának kiküszöbölésére alkották meg az elméleti közép-Napot, melynek két egymásutáni delelése között eltelt időköze a közép-nap, értéke 24 óra.

A program alapja a naptárelv algoritmizálása. Rendeljünk hozzá minden naphoz egy 0-6 közötti értéket*, s jelöljük i-vel.

Amennyiben egy bizonyos dátumról tudjuk, hogy a dátumhoz rendelt jelölő szám i, akkor bármely más dátum jelölő száma:

$$j \equiv K + i \qquad (\text{mod } 7),$$

ahol a K a két dátum között eltelt napok száma, mely a következő képlettel számítható:

$$K = 365 \cdot e + \sum_{k=1}^{h} N_k + n + k$$

ahol e = a két dátum közötti évek száma
h = az években ki nem fejezhető maradék hónapok száma
n = az években és hónapokban ki nem fejezhető maradék napok száma
N_k = a k-ik $(1 \leqslant k \leqslant h)$ hónapban lévő napok száma
k = egy korrekciós faktor

Igy a fenti kifejezés a következőre módosul:

$$j \equiv e + \sum_{k=1}^{h} N_k + n + k \qquad (\text{mod } 7).$$

A k korrekciós faktor a szökőévek beszámítására szolgál.
A további feladat tehát annak az algoritmusnak előállítása, mely a különböző tagok (mod 7) vett maradékait szolgáltatja.
Az alábbi algol program ismerteti ezt az eljárást.
A melléklet alapján az algoritmus helyessége könnyen ellenőrizhető is.

```
procedure       ÖRÖKNAPTÁR                          (A, B, C, D, E);
      value     A, B, C, D;                         integer A, B, C, D, E;
comment:        A az évszázad, B az évszám második két karaktere, C a hónap sorszáma, D a
                nap száma, E tartalmazza a jelölő számot;
begin integer   A1, M, X, C1, C2, C3, P, Z, H, Y, W;

comment:        évszázad maradéka;

                W: = 7;
                A1: = A – /A ÷ 4/ * 4;
                if A1 = 0 then W: = 0 else W: = W – A1;
```

*A kialakított programban a hozzárendelés a következő:

Szombat	=	0	Szerda	=	4
Vasárnap	=	1	Csütörtök	=	5
Hétfő	=	2	Péntek	=	6
Kedd	=	3			

```
comment:   év maradéka;

           A1: = B – /B ÷ 28/ * 28;
           M:  = A1 ÷ 4;
           M:  = M + A1;
           M:  = M – /M ÷ 7/ * 7;

comment:   nap maradéka;

           X:  = D – /D ÷ 7/ * 7;

comment:   hónap maradéka;

           if C ≤ 2 then C1: = C + 12 else C1: = C;
           C1: = C1 – 3;
           C2: = C1 ÷ 2;
           C3: = C1 – C2 * 2;

if C3 = 0 then P: = 0 else begin
   if C1 ≤ 6 then P: = 3 else begin
   if C = 2 then P: = 3 else P: = 2 end, end;
if C ≤ 9 then P: = P + 1;

A1: = C2 * 5;
A1: = A1 + 3;
A1: = A1 + P;

If C ≤ 2 and E – /B ÷ 4/ * 4 = 0 then
begin A1: = A1 – 1; if B = 0 and A – /A ÷ 4/ * 4 ≠ 0
then A1: = A1 + 1; end;

Z: = A1 – /A1 ÷ 7/ * 7;
H = W + M + X + Z + 14;
Y = A * 10^6 + B * 10^4 + C * 10^2 + D;

if Y ≤ 15821005 and Y ≥ 03250325 then H: = H + 2;

if Y ≤ 03250321 and Y ≥ 00000001 then H: = H + 4;

if A < 15 then begin A1: = 15 – A; H: = H – A1;
   A1: = A1 ÷ 4; H: = H + A1 end;
   E:  = H – /H ÷ 7/ * 7;
end;
```

MELLÉKLET

Január 1. napja időszámításunk kezdetétől napjainkig

Vasárnap	Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat
0001	0030	0020	0060	0010	0050	0040
0080		0070				0090
0130	0120		0110	0100	0140	
	0170	0160		0150	0190	0180
0220		0210	0200	0240		0230
0270	0260		0250	0290	0280	
	0310	0300	0330			0320
0350	0340	0380		0370	0360	
	0390					
0440		0430	0420		0410	0400
0490	0480		0470	0460		0450
	0530	0520		0510	0500	0540
0580		0570	0560		0550	0590
0630	0620		0610	0600	0640	
	0670	0660		0650	0690	0680
0720		0710	0700	0740		0730
0770	0760		0750	0790	0780	
	0810	0800	0840		0830	0820
0860		0850	0890	0880		0870
0910	0900	0940		0930	0920	
	0950	0990	0980		0970	0960
1000	1040		1030	1020		1010
1050	1090	1080		1070	1060	
1140		1130	1120		1110	1100
1190	1180		1170	1160		1150
	1230	1220		1210	1200	1240
1280		1270	1260		1250	1290
1330	1320		1310	1300	1340	
	1370	1360		1350	1390	1380
1420		1410	1400	1440		1430
1470	1460		1450	1490	1480	
	1510	1500	1540		1530	1520
1560	1590	1550		1580		1570

Vasárnap	Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat
1640		1630	1620		1610	1600
1690	1680		1670	1660		1650
1730	1720		1710		1700	
	1770	1760		1750	1740	1780
					1790	
	1810		1800			
1860		1850	1840		1830	1820
			1890	1880		1870
	1900	1930	1920			1910
1950	1940	1980		1970	1960	
	1990					
			2020		2010	2000

Beérkezett: 1972 június 25.

S u m m a r y

Perpetual calendar

In everyday life we often meet tasks connected with the perpetual calendar principle. For the calendar principle appearing in the form of a calendar an algorithm can easily be constructed so that it can be computerized. The paper presents an ALGOL program for this.

Р е з ю м е

Вечный календарь

В практической жизни встречаются такие задачи, которые могут быть приведены к принципу вечного календаря. Принцип календаря появляющийся в форме календаря можно легко алгоритмировать и в следствие этого решать на ЦВМ. В качестве примера приведена программа решения задачи на языке АЛГОЛ.

TARTALOMJEGYZÉK