

Lengyelforma

A feladat

... megfogalmazását lásd a [lengyel.fel](#) fájlban:

```
1  (*
2  Feladat:
3      Aritmetikai kifejezés kiértékelése.
4
5  A. Részletesebben:
6  A1. Egy kif sztringben hagyományos infix kifejezés van.
7      Ezt lexikális egységeire bontjuk (lexEgys): lexKif.
8      A lexEgys-eket hagyományos Text-ben (lexKif) tároljuk:
9      egy sor - egy lexikális egység bontásban.
10 A2. Az előbbi lexikális egységeket tartalmazó file-ból
11 kiindulva előállítjuk a post-fix lengyelformáját: pfKif.
12 Ez is egy Text (pfKif), előbbivel azonos konvencióval, azaz
13 egy sor - egy lexikális egység.
14 A3. A pfKif file-ból továbblépve kiszámoljuk a kifejezés
15 értékét.
16 Ha voltak a kifejezésben változók, akkor e programnak kell
17 gondoskodnia az értékek beolvasásáról is.
18
19 B. Még részletesebben:
20 B1. Megengedett lexikális egységek. Azaz, amiket tartalmazhat
21 a bemenő (kif) sztring:
22 B1.1. Számkonstansok
23 B1.2. Számváltozók (azonosítója egy betüből álló szöveg)
24 B1.3. Elemi aritmetikai műveleti jelek ('+', '-', '*', '/', '^')
25 B1.4. Néhány valósértékű aritmetikai függvény ('Sin', 'Cos',
26 'Sqrt', 'Tg', 'Pi'...)
27 B2. A bemenő kifejezéssel szemben elvárt speciális feltételek:
28 B2.1. Minden lexikális egység között pontosan egy szóköz van.
29 B2.2. A függvények argumentumát kötelező bezárójelezni.
30 Pl.:
31     a * Sin ( b * x + Pi / 2 ) - 5.4321
32 B2.3. Kis- és nagybetűket nem különböztetünk meg.
33 B2.4. A kifejezés előtt és után nincsen szóköz.
34 B3. A lexikális egységek megjelenése a lexKif Text-ben: értékpár
35 formájában.
36     lexEgys --> (miféle + értéke)
37 A miféle és hozzárendelt kódja:
38 * Számkonstans: 0
39 * Számváltozó: 1
40 * Művelet (és a zárójelek): 2
41 Egy néhány lexikális egységre példa:
42     ( 0 , '5.4321' )
43     ( 1 , 'a' )
44     ( 2 , 'Sin' )
45 A lexKif file-ra példa:
46     ( 1 , 'a' )
47     ( 2 , '*' )
48     ( 2 , 'Sin' )
```

```

49      ( 2 , '(' )
50      ( 1 , 'b' )
51      ( 2 , '*' )
52      ( 1 , 'x' )
53      ( 2 , '+' )
54      ...
55      ( 0 , '5.4321' )

```

B4. A lexikális egységek megjelenése a pfKif Text-ben: értékpár formájában.

lexEgys --> (miféle + értéke)

A pfKif file-ra példa:

```

60      ( 1 , 'a' )
61      ( 1 , 'b' )
62      ( 1 , 'x' )
63      ( 2 , '*' )
64      ( 2 , 'Pi' )
65      ( 0 , '2' )
66      ( 2 , '/' )
67      ( 2 , '+' )
68      ( 2 , 'Sin' )
69      ( 2 , '*' )
70      ...

```

C. Az egyes részfeladatokat (l. A1-A3.) önállóan futó programok végzik el. A kapcsolatot file-okon keresztül tartják. (Elképzelhető, hogy a fent nevesített file-okon túl [lexKif, pfKif] továbbiak is célszerűek [pl. a műveleti táblát tartalmazó file, ami soronként felsorolja a legális operátorokat esetleg precedenciájukkal együtt; eképpen:

```

77      + 5
78      - 5
79      * 6
80      / 6
81      ^ 7
82      Pi 10
83      Sin 10
84      Cos 10
85      Tg 10
86      Sqrt 10
87      )

```

*)

Az A1 részfeladat megoldása

A lengyelformával kapcsolatos dolgokat, amelyeket az egyes részfeladatok programjai egyformán kell ismerjenek a [lengyel.inc](#)-ben található:

```

1  (*
2  Lengyel.inc: Lengyelforma kiértékelés programjainak közös dolgai
3  *)
4  Type TFajta=(Szam, Valt, Muv, Vege);
5  Const SFajta:Array [TFajta] of String=('Szam', 'Valt', 'Muv', 'Vege');
6  KFajta:Array [TFajta] of Byte=(0,1,2, Ord(Vege));
7  Type TMuv=Record
8      jel:String;
9      prec:Byte
10 End;

```

```

11  Const MaxMuv=12;
12  Type TMuvTabla=Array [1..MaxMuv] of TMuv;
13  Const EgyJeluek=7; {ennyi egy jelből álló operátor van,
14                        amelyek az Op elején találhatók}
15      Op:TMuvTabla=((jel:'(';prec:0),(jel:')';prec:0),
16                    (jel:'+';prec:5),(jel:'-';prec:5),(jel:'*';prec:6),
17                    (jel:'/';prec:6),(jel:'^';prec:7),
18                    {innentől vannak a többjeles operátorok/fv.-ek:}
19                    (jel:'PI';prec:10),(jel:'SIN';prec:10),
20                    (jel:'COS';prec:10),(jel:'TG';prec:10),
21                    (jel:'SQRT';prec:10));
22  Type OLexEgys=Object
23      mifele:TFajta;
24      ertek:String;
25      Constructor Letrehoz(mi:TFajta; er:String);
26      Destructor Lerombol;
27      Procedure Readln(Var f:Text);
28      Procedure Writeln(Var f:Text);
29      End;
30      Constructor OLexEgys.Letrehoz(mi:TFajta; er:String);
31      Begin
32          mifele:=mi; ertek:=er;
33      End;
34      Destructor OLexEgys.Lerombol;
35      Begin
36      End;
37      Procedure OLexEgys.Readln(Var f:Text);
38          Var kod:Byte;
39      Begin {Ef: a file-ban a lexEgys 1 sorban értékpár.
40              1. érték: számkód, 2. érték: szöveg}
41          System.Readln(f,kod,ertek);
42          mifele:=TFajta(kod);
43      End;
44      Procedure OLexEgys.Writeln(Var f:Text);
45      Begin
46          System.Writeln(f,KFajta[mifele], ' ',ertek);
47      End;
48

```

A lexikális elemekre bontás megoldása ([lexKiert.pas](#)):

```

Program LexikalisKiertekeses; {az A1. reszfeladat programja}
(*
... 1. fent a lengyel.fel fájlban
*)
91  Uses Newdelay,Crt,Crt22;
92
93  {\$i AltRutin.inc}
94  {\$i lengyel.inc -- a lengyelforma kiértékelés programjainak közös dolgai}
95
96  Var   kif:String; {billentyű input}
97        lexKif:Text; {file output: Text-esített,
98                      soregységre tördelt OLexEgys file}
99  Const lexKifFn='lexKif.txt';
100
101  Const tollSzin=Black;
102        papirSzin=White;

```

```

103         link:=128;
104
105     Function Beolvasas:String;
106     Var i,j:Byte;
107     ss,s:String;
108 Begin
109     UjLap('Lengyelforma -- kifejezés-beolvasás',0);
110     Write('Kérem az aritmetikai kifejezést:');
111     Readln(s);
112     {nagybetűsit;}
113     For i:=1 to Length(s) do s[i]:=UpCase(s[i]);
114
115     {a műveleti jelek szóközzel tagolása --ha még nem lenne--;}
116     For i:=1 to MaxMuv do
117     Begin
118         ss:='';
119         Repeat
120             j:=Pos(Op[i].jel,s);
121             If j>0 then
122                 Begin
123                     ss:=ss+Copy(s,1,j-1)+' '+Copy(s,j,Length(Op[i].jel))+' ';
124                     s:=Copy(s,j+Length(Op[i].jel),Length(s)-j-Length(Op[i].jel)+1)
125                 End;
126             Until j=0;
127             s:=ss+s
128         End;
129     {elő- és végszóközök levágása;}
130     i:=1;
131     While (i>0) and (s[i]=' ') do Inc(i);
132     Delete(s,1,i-1);
133     i:=Length(s);
134     While (i>0) and (s[i]=' ') do Dec(i);
135     Delete(s,i+1,Length(s)-i);
136     {fölösleges szóközök elhagyása;}
137     i:=1; {bármilyen ami >0}
138     While i>0 do
139     Begin
140         i:=Pos(' ',s); Delete(s,i,1);
141     End;
142
143     Beolvasas:=s
144 End;
145
146 Procedure FileMegnyitas(fN:String; Var f:Text);
147     Var hiba:Integer;
148 Begin
149     UjLap('Lengyelforma -- file-megnyitás',0);
150     Repeat
151         {$i-}
152         Assign(f,fN); Reset(f);
153         hiba:=IOResult;
154         If hiba<>0 then
155             Begin
156                 Rewrite(f);
157                 hiba:=IOResult;
158                 If hiba<>0 then
159                     Begin

```

```

160         TextColor(Red+Blink);
161         Writeln('A file nem nyitható meg:');
162         Writeln(' Hibás file-név. ');
163         TextColor(tollSzin);
164         BillreVar
165     End
166 End
167 Else
168 Begin
169     Rewrite(f);
170     TextColor(Red+Blink);
171     Writeln(' Ilyen file már van. A file felülíródik. ');
172     TextColor(tollSzin);
173     BillreVar
174 End;
175     {$i+}
176 Until hiba=0;
177 End;
178
179 Function Elemez(le:String):TFajta;
180     Var x:Real;
181         i,siker:Integer;
182 Begin
183     Val(le,x,siker);
184     If siker<>0 then
185     Begin
186         i:=1;
187         While (i<=MaxMuv) and (Op[i].jel<>le) do Inc(i);
188         If i<=MaxMuv then Elemez:=Muv
189             else Elemez:=Valt;
190     End
191     else
192     Begin
193         Elemez:=Szam
194     End;
195 End;
196
197 Procedure Felbont(kif:String; Var lk:Text);
198     {Ef: Minden lexikális egység között pontosan egy szóköz van.}
199     Var miASzosz:TFajta;
200         ig:Byte;
201         le:String;
202         leO:OLexEgys;
203 Begin
204     UjLap('Lengyelforma -- kifejezés-felbontás',0);
205     ig:=Length(kif);
206     While ig>0 do
207     Begin
208         ig:=Pos(' ',kif);
209         If ig>0 then
210         Begin
211             le:=Copy(kif,1,ig-1);
212             kif:=Copy(kif,ig+1,Length(kif)-ig);
213             miASzosz:=Elemez(le);
214         End
215         else

```

```

216      Begin
217          le:=kif;
218          kif:='';
219          miASzosz:=Elemesz(le);
220      End;
221      leO.Letrehoz(miASzosz,le);           A def. lásd előbb.
222      leO.Writeln(lk);
223      leO.Lerombol;
224      {ellenőrző kiírás:}
225      Writeln(KFajta[miASzosz], ' (' ,SFajta[miASzosz],') ',le);
226  End;
227  End;
228
229  Begin
230      UjLap('Lengyelforma',0);
231      TextColor(tollSzin); TextBackground(papirSzin);
232      kif:=Beolvasas;
233      FileMegnyitas(lexKifFn,lexKif);
234      Felbont(kif,lexKif);
235      Close(lexKif);
236  End.

```

Minden hozzávaló letöltése: [lengyel.zip](#).