

5. Operátorok és kifejezések

Írjunk függvényt, amely az időt másodpercben olvassa be és átszámítja óra, perc és mpercbe! (*oravaltas*)

```
program oravaltas;
var
  ido, mperc, perc, ora: longint;
begin
  writeln('Kérek egy egész számot!');
  write('Idő [mperc]: '); readln(ido);
  ora := ido div 3600;
  mperc:= ido mod 60;
  perc:= ido mod 3600 div 60;
  writeln(ora, ' óra ',perc, ' perc ',mperc,' mperc');
  readln;
end.
```

A program futásának eredménye:

```
Kérek egy egész számot!
Idő [mperc]: 7275
2 óra 1 perc 15 mperc
```

Írjunk programot, amely beolvassa a derékszögű háromszög két befogóját és kiszámítja a derékszögű háromszög kerületét és területét! (*dszh*)

```
program dszh;
var
  befogol, befogo2, atfogo, ker, ter: real;
begin
  writeln('Derékszögű háromszög befogói');
  write('1. befogó: '); readln(befogol);
  write('2. befogó: '); readln(befogo2);
  atfogo:= sqrt(sqr(befogol)+sqr(befogo2));
  ker:= befogol+befogo2+atfogo;
  ter:= befogol*befogo2/2;
  writeln;
  writeln('Átfogó: ',atfogo:6:1);
  writeln('Kerület: ',ker:6:1);
  writeln('Terület: ',ter:6:1);
  readln;
end.
```

A program futásának eredménye:

```
Derékszögű háromszög befogói
1. befogó: 3
2. befogó: 4
```

5. MEGOLDÁSOK

Átfogó: 5.0
Kerület: 12.0
Terület: 6.0

Írjunk programot, amely beolvas egy pozitív, valós számot és számítsuk ki a szám négyzetét és a négyzetgyökét, az eredményt lebegőpontos és fixpontos alakban is jelenítsük meg! (*szamol*)

```
program szamol;
var
  x,hatvany,gyok : real;
begin
  write('Szám:'); readln(x);
  hatvany:=sqr(x);
  gyok:=sqrt(x);
  writeln(' A szám második hatványa: ',hatvany);
  writeln(' A szám négyzetgyöke:', gyok);
  writeln(' A szám második hatványa:',hatvany:6:2);
  writeln(' A szám négyzetgyöke:', gyok:8:3);
  readln;
end.
```

A program futásának eredménye:

```
Szám:16
A szám második hatványa: 2.560000000000000E+002
A szám négyzetgyöke: 4.000000000000000E+000
A szám második hatványa:256.00
A szám négyzetgyöke: 4.000
```

Értékeljük ki az alábbi programot az *inc()*, *dec()*, *pred()* és a *succ()* gyakorlására! (*fg1*)

```
program fg1;
var
  a,b:integer;
begin
  a:= 2; b:= 4;
  writeln('a: ',a,' b: ',b);
  Dec(a); Inc(b);
  writeln('Dec csökkent a: ',a);
  writeln('Inc növel b: ',b);
  writeln('Pred előző b: ',Pred(b));
  writeln('Succ következő a: ',Succ(a));
  readln;
end.
```

A program futásának eredménye:

```
a: 2 b: 4
Dec csökkent a: 1
Inc növel b: 5
Pred előző b: 4
Succ következő a: 2
```

Értékeljük ki az alábbi programot az *abs()*, *exp()*, *ln()*, *random()* függvények és a *randomize* eljárás használatára! (fg2)

```

program fg2;
var
  a,b,c:real;
begin
  a:= -2.13; b:= 4.56;
  writeln('a: ',a,' b: ',b);
  writeln('abs(a) : ',abs(a));
  writeln('Harmadik hatvány b: ',exp(ln(b)*3):6:2);
  writeln('Harmadik gyöke b: ',exp(ln(b)/3):6:2);
  randomize;
  write('Véletlenszám határral ');
  writeln('megadva 0-89 között: ',random(90));
  writeln('Véletlenszám: ',random);
  readln;
end.

```

A program futásának eredménye:

```

a: -2.130000000000000E+000  b:  4.560000000000000E+000
abs(a) :  2.129999999999999E+000
Harmadik hatvány b:  94.82
Harmadik gyöke b:  1.66
Véletlenszám határral megadva 0-89 között: 82
Véletlenszám:  6.5980083914473653E-0001

```

Értékeljük ki az alábbi programot, mi jelenik meg a képernyőn! (kiertekel)

```

program kiertekel;
{ kifejezések típusa }
begin
  writeln('egész: 7-4      = ', 7-4);
  writeln('valós: 3.0*9    = ', 3.0*9: 6:2);
  writeln('egész: 17 mod 7 = ', 17 mod 7);
  writeln('valós: 9/3      = ', 9/3:4:1);
  writeln('egész: 29 div 5 = ', 29 div 5);
  writeln('valós: 12.4/2   = ', 12.4/2:6:2);
  writeln('hibás: 5.2 div 5');
  writeln('hibás: 6 mod 3.0');
  readln;
end.

```

A program futásának eredménye:

```

egész: 7-4      = 3
valós: 3.0*9    = 27.00
egész: 17 mod 7 = 3
valós: 9/3      = 3.0
egész: 29 div 5 = 5
valós: 12.4/2   = 6.20
hibás: 5.2 div 5
hibás: 6 mod 3.0

```