

Számítógép program: Egy utasítási sorozat a gép nyelvére fordítva
(hasonló a recepthez)

Program: - szekvencia (egymás utáni parancsok)
- szelekció (elöntés)
- iteráció (ciklus - ismétlés)

Változó: Olyan érték, amely a program futtatása során változhat.

(Amelyik nem változik, azt konstansnak nevezjük.)

Meg kell adni, hogy milyen típusú legyen (pl: $x \in \dots$),
mert ettől függ, hogy milyen művelet végezhető vele.

Pascal program:

(belépés: C:\ turbo ← vagy C:\ bp ←)

- 1, deklarációs rész (érteladás)
- 2, főprogram (utasítások)

2, kezdés: begin

befejezés: end.

Az utasítások végén ; → a program zárása

A program futtatása: RUN (menüsorban) ill. ctrl+F9

A program végeredményének megtekintése: alt+F5

képernyőre kiírás: WRITE('')

[:illetve, ha szöveges üzenet kerül a megjeleníteni kívántak, -vel választjuk el egymástól.]

1 sorral lejjebb írja ki a program a képernyőre: WRITELN('')

új változó létrehozása: VAR, változónev: típus

Program: Igaz-e a Gexa nevet úgy, hogy a változó a név legyen, és ehhez rendeljük a "Gexa"-t.

```
MO: var nev:string;
begin
  nev := 'Gexa';
  writeln (nev);
end.
```

SZÖVEGES változók: STRING

beolvasás (+ beolvasás 1 sor emelésekkel): READ (); + READLN ()

NUMERIKUS változók:

számítógép: BYTE (1 byte - 256 tárolható: 0... 255)

WORD (2 byte - 65536 -1 - : 0... 65535)

SHORTINT (1 byte - 256 -1 - : -128... 127)

INTEGER (2 byte - 65536 -1 - : -32768... 32767)

LONGINT (4 byte - 65000^2 -1 - : kb. 2 milliárd ... 2 milliárd)

REAL (6 byte → lebegőpontos ábrázolás)

alapvető műveleti jelek: +; -; *; /

lehet: $a := a + 1$ ⇒ előbb a jobb oldalt számolja ki, utána ad értéket!

Program: Egy 100-nál kisebb szám köbet hozza-e a gép!

```
MO: var a := byte;
kol: 1
begin
  readln (a);
  a := a * a * a;
  writeln (a);
end.
```


Program: Olvassuk be egy h-ű Δ-ner a 3 oldalát és adjuk meg a kerületét és a területét!

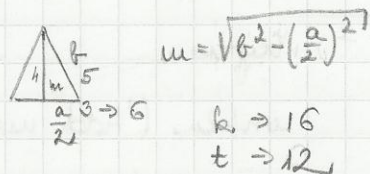
```

MO:   var a: real;
      deszkópas var b: real;
      var c: real;
      begin
        writeln('Add meg az egyik befogót!');
        readln(a);
        writeln('Add meg a másik befogót!');
        readln(b);
        writeln('Add meg az átfogót');
        readln(c);
        writeln('A kerület:', a+b+c, '!');
        writeln('A terület:', a*b/2, '!');
      end.
  
```

write (a : 2 : 0)
 v.kis ? hely a töredékes
 kicátásca száma
 szám

pl.: 3,146 (a: 3: 2) → 3,14
 3,1245 (a: 0: 2) → 3,1
 a gép ilyenkor kört!

Program: Egyszögű három Δ 2 oldalát beolvassuk, és kiszámoljuk a kerületét és a területét.



sqrt (u) → u gyöket adja meg.

```

MO:   var a, b, k, t, m: real;
      begin
        writeln('Add meg a Δ oldalait!');
        readln(b);
  
```



```

writeln ('Add meg a  $\Delta$  alapját!');
readln (a);
k := a + b + c;
writeln (' $\Delta$  terület:', k:3:1, '!');
m := sqrt (b*b - a/2 * a/2);
t := a*m/2;
writeln (' $\Delta$  területc:', t:3:1, '!');
end.

```

mentve: adriana.pas -nem!

IF \langle feltétel \rangle \rightarrow v. mi feltételt kell bírni
 then \langle utasítás 1 \rangle
 [else \langle utasítás 2 \rangle] \rightarrow nem kell bírni

ha el akarom dönteni, hogy $a > 5 \Rightarrow$

if $a > 5$ then writeln ('a nagyobb 5') \rightarrow nincs pontosvessző

```

else begin
  a := a + 5;
  writeln ('most már nagyobb');
end;

```

PROGRAM

* program vizsgálja meg, hogy ne legyen nagy az alap.

MO:

egyenk. pas

```

var a, b, k, t, m: real;
begin
  writeln ('Add meg a  $\Delta$  oldalait!');
  readln (b);
  writeln ('Add meg a  $\Delta$  alapját!');
  readln (a);
  if  $a \geq 2 * b$  then writeln ('Túl nagy alap!')
  else begin

```



```

k := a + b + b;
writeln('A terület:', k:3:1, '!');
m := sqrt(b * b - a / 2 * a / 2);
t := a * m / 2;
writeln('A terület:', t:3:1, '!');
end;

```

end.

Open: F3

save: F2

exit: alt + X

User screen: alt + F5

PROGRAM: Kéjűre be 1 derékszögű Δ oldalait, és az területét és a
Csere-pas területét számolja ki!

a	x := a;	a := a + b;
b	a := b;	b := a - b;
x	b := x;	a := a - b;

a = 3	c = 5	a = 4
b = 4	b = 4	b = 5
c = 5	a = 3	c = 3
t = 6		
k = 12		

```

MO :
var a, b, c, x: real;
begin
writeln('Add + a  $\Delta$  csúcs oldalait!');
readln(a);

```



```

writeln('Add weg a ma'rat u!');
readln(b);
writeln('Add weg a 3.-at is!');
readln(c);
if a > c then begin
    x := a;
    a := c;
    c := x;
end;
if b > c then begin
    x := b;
    b := c;
    c := x;
end;
writeln('A lenitet', a + b + c, '!');
writeln('A t'itet:', a * b / 2, '!');
end.

```

PROG: Bedvas 3 x'not, is eija az atlagat.

MO: atlag.pas var a, b, c: real;

```

begin
writeln('Add + az 1. x'not!');
readln(a);
writeln('Add + a 2.-at!');
readln(b);
writeln('Add + a 3.-at!');
readln(c);
writeln('Az atlag:', (a + b + c) / 3:3:1, '!');
end.

```


uses crt; → clrscr - Égyenlítő
 kezdés után

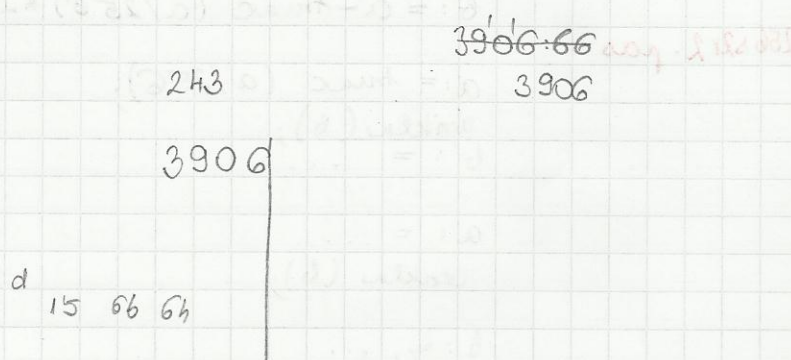
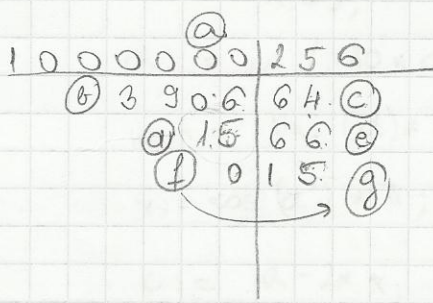
real → integer

trunc → $d := \text{trunc}((a+b+c)/3);$
round → $d := \text{round}((a+b+c)/3)$

egész számok → 3,8-nál 3-at kapunk egész rész
 kerekítés '1/2-től felfelé' → 3,8-nál 4-et kapunk

int: az egész részt adja meg
 real-ből → kettő szíval

PROG: 1000000-s értéket kell a programmal eltalálni 256-os számrendszerben!



```

MEGO:  USES CRT;
256 sz. pas  VAR a: real;
        var b, c, d, e: real;
        begin
            clrscr;
            writeln('Adj meg 1 számot max 1 millióig!');
            readln(a);
            if a > 1000000 then writeln('Túl nagy a szám!')
            else
                begin
                    b := trunc(a / 256);
                    c := a - (b * 256);
                    if b < 256 then writeln(b:3:0, c:3:0)
                    else

```