

8 A WHILE..DO ciklus

- a **WHILE ... DO** ciklus

8.1 A WHILE ... DO (előtesztelési ciklus)

Ez a ciklus a **for** ciklushoz hasonlóan megismétel néhány parancsot többször egymás után. A különbség abban van, hogy míg a **for** ciklusnál a ciklusváltozó kezdő- és végértéke határozta meg az ismétlések számát (pl. **for i:=1 to 8** ciklusnál az adott parancsot nyolcszor ismételte meg), a **while..do** ciklusnál az ismétlések számát nem egy ciklusváltozó, hanem egy feltétel határozza meg (pl. **a<b**). Amíg a feltétel igaz, addig az adott parancsokat a **while..do** ciklus ismételni fogja. A ciklus formája:

```
while feltétel do parancs;
```

vagy ha több parancsot szeretnénk elvégezni a cikluson belül, akkor hasonlóan mint a **for** ciklusnál és az **if** feltételvizsgálatnál, használjuk a **begin..end** utasításokat:

```
while feltétel do begin
    parancs1;
    parancs2;
    ...
    parancsN;
end;
```

Magyarul: amíg a feltétel igaz, ismételd a ciklusban levő parancsokat.

A feltétel itt is hasonlóan mint az **if** feltételvizsgálatnál lehet egy vagy több kifejezésből is, melyeknél használhatjuk az **AND** (és), **OR** (vagy), **NOT** (nem), **XOR** (kizáró vagy) műveleteket.

A ciklus a következő képen működik: a számítógép megnézi, hogy a feltétel igaz-e.

- Ha nem igaz, akkor a ciklusban levő parancsokat nem hajtja végre egyszer sem, hanem a program folytatódik a ciklus utáni utasításokkal.
- Ha a feltétel igaz, végrehajtja a ciklusban levő parancsokat. Ismét megvizsgálja a feltételt, amely ha még mindig igaz, akkor ismét végrehajtja a ciklusban levő parancsokat. Majd ismét megvizsgálja a feltételt, amely ha meg mindig igaz, akkor ismét végrehajtja a parancsokat... Ha a feltétel már nem igaz, akkor a parancsokat nem hajtja végre, hanem folytatódik a program futása a ciklus utáni utasításokkal.

Feladat: Készítsünk programot az **n!** kiszámítására a **for** ciklus használata nélkül (a **while** ciklus segítségével).

Megoldás: A programunk a következő képen néz ki:

```
program Pelda23;
uses crt;
var n,szorzat:integer;
begin
clrscr;
write('Add meg az N erteket: ');
readln(n);
szorzat:=1;
while n>1 do begin
    szorzat:=szorzat*n;
    dec(n); { csokkenti n erteket 1-gyel,
    irhattuk volna helyette n:=n-1-et is }
end;
writeln('N! = ',szorzat);
readln;
end.
```

Ha programon belül megjegyzést szeretnénk hozzáfűzni valamihez, akkor { } jelek között tehetjük meg.

Feladatok:

1. Készítsünk programot, amely beolvasson egy egész számot, majd elosztja 2-vel annyiszor, ahányszor lehet és közben felírja a számot a kettes számok szorzataként megszorozva egy olyan számmal, amely már nem osztható 2-vel.

Ha a szám egyszer sem osztható kettővel, akkor:

Ahhoz, hogy a szám osztható-e kettővel használjuk a **mod** függvényt. Pl. X akkor osztható 2-vel, ha **X mod 2 = 0**. A szám elosztását a **div** függvénnyel végezzük el. **(while01)**

```
Kerek egy egész számot: 120
120 = 2*2*2*15
```

```
Kerek egy egész számot: 17
17 = 17
```

2. Készítsünk programot, amely bekér egy egész számot, majd mindaddig kér be további egész számokat, amíg nem adjuk meg a 0-t. A program határozza meg és írja ki a beadott egész számok közül a legnagyobbat. **(while02)**

```
Kerek egy számot: 23
Kerek egy számot: 56
Kerek egy számot: 16
Kerek egy számot: 32
Kerek egy számot: 0
-----
A legnagyobb szám: 56
```

3. Készítsünk programot, amely bekér egy n egész számot, majd kiírjuk az első „n” szám összegét. (while ciklussal) **(while03)**

```
Adj meg egy számot: 5
-----
Összeg: 15
```

4. Készítsünk programot, amely bekér két számot 0 és 20 között, majd írasson ki annyi „X”-et, amennyi a két szám különbsége. (while) **(while04)**

```
Kerem az első számot: 17
Kerem a második számot: 13
-----
XXXX
```

```
Kerem az első számot: 23
A szám nem 0 és 20 közötti!
Kerem az első számot:
```

5. Készítsünk programot, amellyel bekérünk egy „n” számot. Majd kiírjuk az „n” szám szorzatait 100-ig. (Az „n” egy 0 és 10 közötti szám) **(while05)**

```
Kerem az „n” számot: 8
-----
8; 16;24;32;40;48;56;64;72;80;88;96;
```

```
Kerem az „n” számot: 23
A szám nem 0 és 10 közötti!
Kerem újra az „n” számot!
```