

Az IF feltételvizsgálat

az IF ... THEN parancs

- a feltétel ELSE ága

Az IF ... THEN parancs

Gyakran előfordul, hogy programunkban valamilyen feltételtől függően szeretnénk parancsot végrehajtani. Például akkor szeretnénk végrehajtani valamilyen utasítást, ha az *i* változó értéke nagyobb mint **5**. Erre általánosan a következő utasítás szolgál:

```
if feltétel then parancs;
```

A mi példánkban (ha *i* nagyobb mint 5) ez így nézne ki:

```
if i>5 then parancs;
```

A feltételben bármilyen két kifejezést összehasonlíthatunk. A kifejezések összehasonlítására használhatjuk a <, >, =, <=, >=, <> (nem egyenlő) jeleket, melyek igaz/hamis (true/false) logikai értékeket adnak vissza. Ha ez a logikai érték igaz (true), akkor a parancs végrehajtódik.

A logikai értékeket összekapcsolhatjuk az **AND**(és), **OR** (vagy), **NOT** (nem), **XOR** (kizáró vagy) műveletekkel. Így egyszerre több feltételt is vizsgálhatunk. Ilyenkor a feltételeket, amelyeket összekapcsoljuk, zárójelbe tesszük.

Természetesen itt is, hasonlóan mint a **for** ciklusnál egyszerre több parancsot is végrehajthatunk a feltételen belül. Ekkor a parancsokat a **begin ... end;** közé rakjuk:

```
if feltétel then begin
```

```
  parancs1;
```

```
  parancs2;
```

```
  ...
```

```
end;
```

Feladat: Készítsünk programot, amely bekér három pozitív egész számot, és kiírja őket nagyság szerint növekvő sorrendben!

Megoldás: A programban a három számnak 3 változót fogunk használni (**a**, **b**, **c**). Ezeket cserélgetéssel sorba fogjuk rakni úgy, hogy az **a** változóban legyen a legkisebb szám, a **b**-ben a középső és a **c**-ben a legnagyobb szám. A cserékhez egy **x** segédváltozót használunk.

```
program Pelda19;  
var a,b,c,x:integer;  
begin  
  write('Kerem az elso szamot: ');  
  readln(a);  
  write('Kerem a masodik szamot: ');  
  readln(b);  
  write('Kerem a harmadik szamot: ');  
  readln(c);  
  if a>b then begin  
    x:=a;  
    a:=b;  
    b:=x;  
  end;  
  if a>c then begin  
    x:=a;  
    a:=c;  
    c:=x;  
  end;  
  if b>c then begin  
    x:=b;
```

```

b:=c;
c:=x;
end;
write('A három szám növekvő sorrendben: ');
writeLn(a, ' < ', b, ' < ', c);
end.

```

A programunkban használtunk egy új függvényt, az **inc()** függvényt (increase). Ez a függvény növeli a változó értékét 1-gyel, tehát az **inc(nagy)** helyett írhattuk volna ezt is: **nagy:=nagy+1**.

Hasonlóan létezik egy függvény, amely csökkenti a változó értékét eggyel, ez a **dec()** függvény (decrease).

A feltétel ELSE ága

A programozás során sokszor előfordul, hogy akkor is végre akarunk hajtani valamilyen utasítást, ha a feltétel nem igaz. Ebben az esetben az **if ... then** parancsunk bővíteni fog egy **else** résszel:

```

if feltétel then parancs1
else parancs2;

```

Ha a feltétel igaz, akkor a *parancs1* hajtódik végre, egyébként a *parancs2* hajtódik végre. Fontos megjegyeznünk, hogy ha az **if** feltételvizsgálatnak van **else** ága, akkor a *parancs1* után nem rakunk pontosvesszőt. Természetesen itt is mindkét parancs helyett tehetünk több egymást követő utasítást is. Ilyenkor minden esetben a **begin ... end** kulcsszavakkal kapcsoljuk össze az egymást követő parancsokat:

```

if feltétel then begin

```

```

parancs1i;
parancs2i;

```

```

...

```

```

end
else begin
parancs1h;
parancs2h;

```

```

...

```

```

end;

```

Logikai műveletek eredményei:

AND	IGAZ	HAMIS
IGAZ	IGAZ	HAMIS
HAMIS	HAMIS	HAMIS

OR	IGAZ	HAMIS
IGAZ	IGAZ	IGAZ
HAMIS	IGAZ	HAMIS