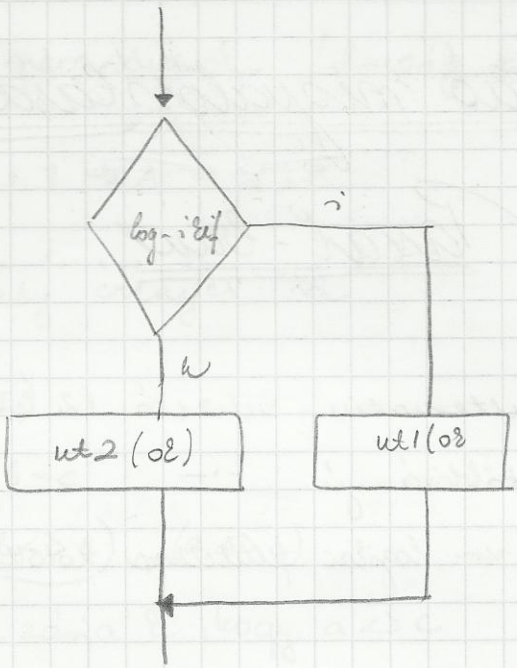


ELSE előtt NINCS pontosvessző!!!

Többirányú lehet úgy is, hogy if-eket egymásba ágyazunk.

/2. Esetvizsgáló többirányú elágazás:

case (selector) of

címke ↖ c1 : ut1(o2);
c2 : ut2(o2);
⋮
cn : utn(o2);
end;

nem fontos a „;".

selector: sorszámozható típusú változó, vagy egy ilyen kifejezés, de mindenesetben sorszámozott típusú változót ad.
Bármilyen egész (+, -), lehet betű is (=karakteres változó) a címkék konstans címkét kapnak (a selector típusával megegyezik)

végrehajtására: a selector értékét a gép megnézi, megkeresi a címkelistában és az annyiadikra lép. pl.: ha az 3, akkor a 3-ra fog ugrani

pl.: readln (sorszám);
case sorszám of
1: write ('elégtelen');
2: write ('elégséges');
:
5: write ('jeles');
end.

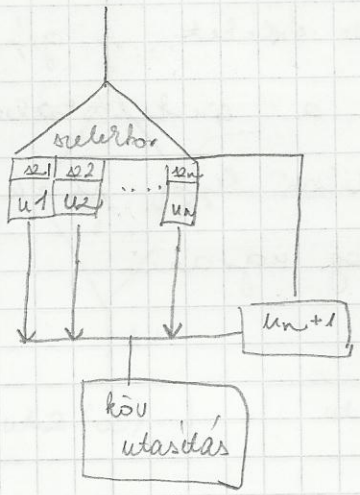
Ha egyik címke sem fedi meg a selector értékét, akkor nem csinál semmit (= üres utasítás)

case (x) of
c1 (of)
c2 (of)
:
else
ut... (f)
end;

A címkelistában lehetnek összetett elemek. És a halmaz v. intervallum

case adat of
1...10: ut1 (of);
12, 17, 19: ut2 (of);
32: ut3 (of)

case honap of
1, 3, 5, 7, 8, 10, 12: max nap:=31
4, 6, 9, 11: max nap:=30
2: if ken szökőév: 29
else 28.
end.



Ha 1 programrészletet (1 v. több utasítás) egymás után Ezer végrehajtani → iteráció. A ilyen utasítás pedig a ciklus utasítás.

- előírt lépésszám:

for <ciklusparaméter> := kezdőérték { to / downto } végérték do <ciklusmag>

(ide nem kell ";")

működése: to, downto → növekvő v. csökkenő sorrendben halad végig.
(ettől függ)

megvizsgálja a ciklusparaméter aktuális értékét, és ha az nem lépett túl a ciklusparaméter végértékén, akkor a ciklusmagot végrehajtja, egyébként nem. És előírtszámú ciklus.

Ha a ciklusmag több utasításból áll, akkor blokkba kell foglalni

- while (log. kif.) do

ciklusmag

az utasításblokkot (begin end) közé kell tenni

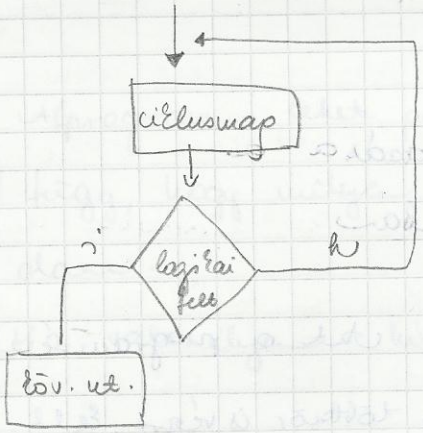
- hátultesztelő hamisra ismétlő ciklus:

repeat

ciklusmag

until (logikai kif)

nem kell begin ... end közé foglalni



FOR ciklus

var a: array [1...50, 1...20] of real;

for i := 1 to 50 do

for j := 1 to 20 do] ciklusmag

a [i; j] := 0

Amikor a ciklusmag maga is ciklus, akkor ciklusok egymásba ágyazásáról beszélünk.

pl.:

Adott egy kétdimenziós tömb és számadatokat tartalmaz.

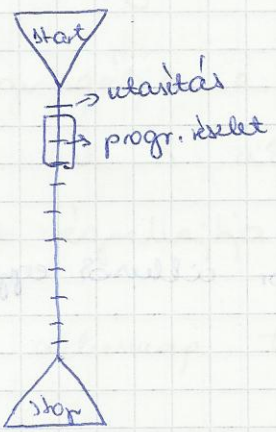
Állapítsuk meg a mátrix sorainak összegét.

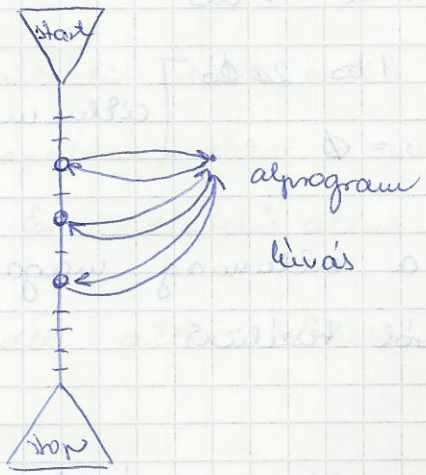
1	2.3	4.7	5.1	6.0
2.9	4.3	12.1	2.0	5.6
7.0	11.5	3.9	4.1	1.8

```
FOR   i:= 1   to   50 do
 begin
 s:= 0
 for j:= 1 to 20 do
 s:= s+a[i,j];
 writeln (s);
 :
 end;
```

Szubrutinok alkalmazása a Pascal programban

Alprogram (szubrutin) fogalma: Azt a programrészletet, amelyet a program futtatása során többször is végre kell hajtani - esetleg változó paraméter értékek mellett - alprogramnak nevezzük. A legtöbb programnyelvben ez a programrészlet önállóan kezelhető és önállóan készíthető.





hívó szegmens

hívott szegmens → a hívó szegmensnek alá van rendelve

Az alregmenüknél is van olyan alregmenü, amiben lehetnek működő utasítások.

```
program név ;
uses crt ;
type ...
const ...
var ...
procedure
function
begin
  :
end.
```

→ A program lehet procedure vagy function, attól függ, hogy milyen problémát akarunk vele megoldani.
→ Ha valamilyen hívó regmens aktivizálja

eljárások (procedures)

procedure név [(form. paraméterlista)] ;
nem kötelező

felépítése szerkezetében teljesen megegyezhet a program felépítésével.

```
[uses ...]
[type ...]
[const ...]
[var ...]
[procedure ...]
 begin
  :
 end;
```

hatáskör: A főprogr. összes deklarációja globális érvényű (érvényesek bármelyik regiemben is.)
Az alprg.-ban deklarált lejegyzések lokális érvényűek (csak ott érvényesek v. az alárendelt ágaiban)

Formális paraméterlista: A par. listán funkcióját tekintve kétféle par. szerepelhet. Bemenő és kimenő paraméter. A bemenő par. értéket visz be az eljárásba, annak végrehajtásakor a hívó szegmensből.
A kimenő par. az eljárás lefutása után értéket ad vissza a hívó regmensnek.

Paraméterátadás módjai:
- globális par. átadás automatikus
- lokális par. átadás: - paraméteres értékszerinti átadása (kiir., vált, tif)
  - paraméteres címszerinti átadása.

Bemenő par-t. mindkét módon át lehet adni.
Kimenő par-t. csak címszerinti lehet átadni.

Az értékes verenter.-n kerül átadásra az érk. szerinti átadásnál.

értér

cím

lokális memóriacímre definiálnak

formális paraméter

aktuális paraméter

A pascal eljárásorientált nyelv.

Függvényeljárás

Mindig van egy és csak egy bemenő paramétere, bemenő opcionális vagy van vagy nincs. Hasonló az eljáráshoz

function név (bemenő paraméter): típus, paraméter típusa

kerekzárójelek megegyeznek az eljárásával, a végrehajtás belépése: a függvény neve egy értékadó utasítás bal oldalán szerepel (legalább 1 helyen)

Heron képlet:

$$T = \sqrt{s \cdot (s-a)(s-b)(s-c)}$$

```
procedure
    Heron (a1, a2, a3:real; var t :real);
  var
      s: real
  begin
      s := (a1+a2+a3)/2;
      t := sqrt (s*(s-a1)*(s-a2)*(s-a3));
      end;
  Heron (3,4,5, ter);
readln (x; y, z);
Heron (x; y, z, ter);
```