

Elemi programozási tételek

1. Az összegzés tétele

Az összegzés tétele összeadja a tömbben lévő számokat. Szorzásra is alkalmas, de ekkor az S 1 kezdőértéket vesz fel.

X: A tömb.

N: A tömb elemszáma, egész.

S: Az eredmény.

Összegzés(N,X,S):

S:=0

Ciklus I=1-től N-ig

S:=S+X(I)

Ciklus vége

Eljárás vége.

...

S:=0;

For i:=1 to N do

S:=S+x[i];

writeln('Az osszeg: ',S);

...

2. Az eldöntés tétele

Az eldöntés tétele megállapítja, hogy a tömbben van-e a feltételnek megfelelő elem.

X: A tömb.

N: A tömb elemszáma, egész.

VAN: Az eredmény, logikai típus.

Eldöntés(N,X,VAN):

I:=1

Ciklus amíg I<=N és nem T(X(I))

I:=I+1

Ciklus vége

VAN:=(I<=N)

Eljárás vége.

...

i:=1;

while (i<=N) and (x[i]<=0) do

i:=i+1;

van:=(I<=N);

if van=true then writeln('Van pozitív szám')

else writeln('Nincs pozitív szám.');

...

Módosítható úgy a feladat, hogy azt döntse el, mindegyik elem megfelelő tulajdonságú-e.

MIND: Az eredmény, logikai típus.

Eldöntés(N,X,MIND):

I:=1

Ciklus amíg I<=N és T(X(I))

I:=I+1

Ciklus vége

MIND:=(I>N)

Eljárás vége.

...

i:=1;

while (i<=N) and (x[i]>0) do

i:=i+1;

mind:=(I>N);

if mind=true then writeln('Van pozitív szám')

else writeln('Nincs pozitív szám.');

...

3. A kiválasztás tétele

Ebben az esetben tudjuk, hogy létezik az adott tulajdonságú elem, a legelső ilyen elem sorszámát kapjuk eredményül.

X: A tömb.

N: A tömb elemszáma, egész.

SORSZ: Az eredmény, egész típus.

Kiválasztás(N,X,SORSZ):

l:=1

Ciklus amíg nem T(X(l))

l:=l+1

Ciklus vége

SORSZ:=l

Eljárás vége.

...

i:=1;

while x[i]<=0 **do**

i:=i+1;

sorsz:=i;

writeln('A legelső pozitív elem sorszáma: ',
sorsz, ' A legelső pozitív elem: ', x[sorsz]);

...

4. Lineáris keresés

A lineáris keresésben megállapítjuk, hogy létezik-e a tömbben az adott tulajdonságú elem, ha igen, megadjuk az elsőt. A tétel az eldöntés és kiválasztás tétel együttese.

X: A tömb.

N: A tömb elemszáma, egész.

VAN: Logikai típus.

SORSZ: Az eredmény, egész típus.

Keresés(N,X,VAN,SORSZ):

l:=1

Ciklus amíg l<=N és nem T(X(l))

l:=l+1

Ciklus vége

VAN:=(l<=N)

Ha VAN **akkor** SORSZ:=l

Eljárás vége.

...

i:=1;

while (i<=N) and (x[i]<=0)

i:=i+1;

van:=(i<=N);

if van=true

then begin

sorsz:=i;

writeln('A pozitív szám: ', x[sorsz])

end

else writeln('Nincs pozitív szám.');

...

5. A megszámolás tétele

A megszámolás tételében megszámoljuk, hogy hány adott tulajdonságú elem van a tömbben.

X: A tömb.

N: A tömb elemszáma, egész.

DB: Az eredmény, egész típus.

Megszámolás(N,X,DB):

DB:=0

Ciklus l=1-től N-ig

Ha T(X(l)) **akkor** DB:=DB+1

Ciklus vége

Eljárás vége.

...

db:=0;

for i:=1 to N **do**

if x[i]>0 **then** db:=db+1;

writeln(db, ' db pozitív szám van.');

...

6. A maximumkiválasztás tétele

A maximumkiválasztás tételében megkeressük az adott tömbben a legnagyobb elemet. Eredményül a sorszámot és magát az értéket kaphatjuk.

X: A tömb.

N: A tömb elemszáma, egész.

MAX: Az eredmény, egész típus.

MAXERT: Az eredmény.

Maximumkiválasztás(N,X,MAX,MAXERT):

MAX:=1: MAXERT:=X(1)

Ciklus I=2-től N-ig

Ha MAXERT<X(I)

akkor MAX:=I : MAXERT:=X(I)

Ciklus vége

Eljárás vége.

...

max:=1; maxert:=x[1];

for i:=2 to N do

if maxert<x[i] then begin

max:=i; maxert:=x[i];

end;

writeln('A legnagyobb elem sorszáma: ',max);

writeln('A legnagyobb elem: ',maxert);

...

7. Egyszerű cserés rendezés

Ezzel az algoritmussal növekvő illetve csökkenő sorrendbe tudom rendezni a tömbben lévő elemeket.

X: A tömb.

N: A tömb elemszáma, egész.

Rendezés(N,X):

Ciklus I=1-től N-1-ig

Ciklus J=I+1-től N-ig

Ha X(I)>X(J)

akkor Csere(X(I),X(J))

Ciklus vége

Ciklus vége

Eljárás vége.

...

for i:=1 to N-1 do

for j:=i+1 to N do

if x[i]>x[j] then begin

s:=x[i];x[i]:=x[j];x[j]:=s;

end;

...

Az algoritmus működési elve:

Végignézi az I ciklusváltozó segítségével az elejétől az utolsó előtti elemig a tömböt. A J ciklusváltozó mindig az adott elemnél nagyobb elemeket veszi sorra a végéig (J:=I+1). Ha a tömbben előbb lévő elem nagyobb, mint a hátrább lévő, akkor cserélni kell.

A csere működési elve:

Ehhez szükségünk egy segédváltozóra. Először az s felvesz az X(I) értékét, ezért ez már nem veszik el. Ezek után az X(I) már felveheti az X(J) értékét, így az sem veszik el. Legvégül az X(J) felveszi az s értékét, ami az X(I). Tehát a csere megtörtént.