

Elem kivétele leírásával:

függvény léptetőből (sor : sor típus; Adat : elem típus): logikai;

vátozás n: egész;

ha nem sorias (sor) akkor

Adat := sor . elem [sor . első]

célitus n := sor . első .. sor . utolsó - 1 ismételt

 sor . elem [i] := sor . elem [i + 1];

c vége;

 sor . utolsó := sor . utolsó - 1;

 léptetőből := igaz

különben

 léptetőből := hamis;

h vége;

vége;

Célitus sor állapotának leírása:

függvény célitásra (sor : sor típus): logikai;

célitásra := (sor . utolsó = \emptyset);

vége;

függvény célisortele (sor : sor típus): logikai;

célisortele := ((sor . első = 1 és sor . utolsó = maxelem) vagy

(sor . utolsó > \emptyset és sor . utolsó = sor . első - 1))

vége;

új elem elhelyezése a célkísorban:

függvény: célkísorba (sor: sor típus; Adat: elem típus): logi Érték;

ha nem célkísor (sor) akkor

ha sor, utolsó < maxelem akkor

sor. utolsó := sor. utolsó + 1

különben

sor. utolsó := 1;

h vége;

sor. elem [sor. utolsó] := Adat;

célkísorba := igaz

különben

célkísorba := hamis;

h vége;

vége;

Elem kivétele a ciliées sorból:

függvény cilsorból (sor: sor típus; Adat: elem típus): logikai;

ha nem cilsorúes (sor) akkor

Adat := sor . elem [sor . első]

ha sor . első = sor . utolsó akkor

sorlexd (sor)

különben

ha sor . első < maxelem akkor

sor . első := sor . első + 1;

különben

sor . első := 1;

u vége;

u vége;

cilsorból := igaz

különben

cilsorból := hamis;

u vége;

vége;

HALMAZ :

deklarálás :

konstans maxelem = maximális - elemxám

tipus elemtipus : tárolandó - elemet - típusa

tipus halmaxtipus : rekord (

elem : tömb [1.. maxelem] elemtipus;

elemxám : egész;

)

inicializálás :

eljárás halmaxrend (halmax : halmaxtipus);

halmax . elemxám := \emptyset ;

vége;

halmax állapotának ellenőrzése:

függvény halmaxüres (halmax : halmaxtipus) : logikai;

halmaxüres := (halmax . elemxám = \emptyset);

vége

függvény halmaxtele (halmax : halmaxtipus) : logikai;

halmaxtele := (halmax . elemxám = maxelem);

vége;

függvény elemc (halmoz: halmoz_típus; Adat: elem_típus):
logikai;

változó i: egész;

i := 1;

amíg (i ≤ halmoz. elemszám) és (halmoz. elem [i] ≠
Adat) ismételt

i := i + 1;

a vége;

elemc := (i = halmoz. elemszám);

vége;

Metszet:

eljárás metszet (A, B, C: halmoz_típus);

változó i: egész;

C. elemszám := 0;

amíg i = 1.. A. elemszám ismételt

ha elemc (B, A. elem [i]) akkor

C. elemszám := C. elemszám + 1;

C. elem [C. elemszám] := A. elem [i];

h vége;

C vége;

vége;

unió:

eljárás unió (A, B, C : halmasztípus);

változó i : egész;

c. elemszám := 1.. A. elemszám ismételt

c. elem [i] := A. elem [i];

c vége;

c. elemszám := A. elemszám;

ciklus $i := 1.. B$. elemszám ismételt

ha nem elem (A, B. elem [i]) akkor

c. elemszám := c. elemszám + 1;

c. elem [c. elemszám] := B. elem [i];

h vége;

c vége;

vége;

különbőség:

eljárás különbőség (A, B, C : halmasztípus);

változó i : egész;

c. elemszám := 0;

ciklus $i := 1.. A$. elemszám ismételt

ha nem elem (B, A. elem [i]) akkor

c. elemszám := c. elemszám + 1;

c. elem [c. elemszám] := A. elem [i];

h vége;

c vége;

vége;

Részhalmozásvizsgálat:

függvény részhalmozás (A, B : halmozástípus): logikai;

változó i : egész;

ha A . elemszám $<$ B . elemszám akkor

részhalmozás := hamis;

különben

$i := 1$;

amíg $(i \leq B$. elemszám) és elem $(A, B$. elem $[i])$ ismét

$i := i + 1$;

a vége;

részhalmozás := $(i > B$. elemszám);

h. vége;

vége;

REKURZIO:

Faktoriális:

függvény fact (n : egész) : valós;

ha $n = 0$ akkor

fact := 1

különben

fact := $n \cdot \text{fact}(n-1)$;

n vége;

vége;

Fibonacci - számok:

függvény fib (n : egész) : valós;

elágazás

amikor $n = 0$:

fib := 0;

amikor $n = 1$:

fib := 1

különben

fib := fib($n-1$) + fib($n-2$);

n vége;

vége;