

Algoritmusok és adatstruktúrák

11. gyakorlat

1. HETI TÉMAKÖR

A mutató típus és a dinamikus tárkezelés. Dinamikus adatstruktúrák: kollekciók, láncolt listák.

2. MEGOLDOTT FELADATOK

Készítsünk teljesen paraméterezett szubrutint (adatszerkezeti táblázat és algoritmus) az alábbi feladatokra!

2.1. Láncolt lista hossza

Egy nagy pozitív egész számot egy egyirányban láncolt lista tárolja úgy, hogy minden láncelem egy db számjegyet tárol. Hány jegyű egy adott nagy szám?

Megoldás: A megoldásra függvényt készítünk, de először a feladathoz illeszkedő láncelem rekordot deklaráljuk. A megoldó szubrutin megkapja a lánc első elemének a mutatóját, ahonnan elindulva egyszerűen végig lépdelünk a lánc elemein, és minden egyes lépésben megnöveljük a kezdetben lenullázott eredményt eggyel. A ciklus befejezése után visszaadjuk az eredményt.

Típus

```
LANCELEM rekord
    SZAMJEGY egész
    KOV LANCELEM rekordra mutató
```

Funkció	Azonosító	Típus	Jelleg
A lista első eleme	ELSO	LANCELEM-re mutató	I
Az eredmény	DB	Egész	M, O
Az aktuális láncelem	AKT	LANCELEM-re mutató	M

```
LANCHOSSZ (ELSO)
AKT<-ELSO
DB<-0
while AKT<>NIL
    AKT<-(*AKT).KOV
    DB<-DB+1
return DB
```

2.2. Nagy szám párossága

Egy nagy pozitív egész számot egy egyirányban láncolt lista tárolja úgy, hogy minden láncelem egy db számjegyet tárol. Páros-e egy adott nagy szám vagy sem? Feltehető, hogy a lánc nem üres.

Megoldás: Az előző feladat megoldásában szereplő láncelem rekordtípust most nem deklaráljuk újra. Hasonlóan végig kell lépdelnünk a nagy szám jegyein (azaz a láncolt lista elemein), de az utolsó elemen meg kell állni, mert attól függ az eredmény értéke. Ha ott páros szám van, akkor a nagy szám is páros, egyébként nem.

Funkció	Azonosító	Típus	Jelleg
---------	-----------	-------	--------

A lista első eleme	ELSO	LANCELEM-re mutató	I
Az aktuális láncelem	AKT	LANCELEM-re mutató	M

```
PAROS (ELSO)
AKT<-ELSO
while (*AKT).KOV<>NIL
    AKT<-(*AKT).KOV
return (*AKT).SZAMJEGY MOD 2=0
```

2.3. Nagy mátrix generálása

Generáljunk véletlenszerűen egy $N \times N$ -es ($N \leq 500$) bináris „nagy” mátrixot! A mátrix egyben (egy statikus tömbbel) nem, de darabolva (esetleg) tárolható a dinamikus memóriában!

Megoldás: A nagy mátrixnak egy kollekciót feleltetünk meg, amellyel sorokra darabolva tárolhatjuk a nagy mátrixot (a dinamikus memóriában). Eljárást készítünk, amelyik a paramétereiben adja vissza az eredményeket. Ha nincs elég szabad memória (hogy a teljes mátrixot tárolni tudjuk), akkor felszabadítjuk a már lefoglalt memóriát.

```
Konstans
/* A sorok, ill. oszlopok maximális száma */
NMAX 500
```

```
Típus
/* A mátrix egy sora */
SOR Egydimenziós Egész típusú adatokból álló tömb[NMAX]
/* A nagy mátrix, mint kollekció */
KOLLEKCIO Egydimenziós SOR típusú adatra mutató mutatókból álló tömb[NMAX]
```

Funkció	Azonosító	Típus	Jelleg
A mátrix mérete	N	Egész	I
A generált mátrix	A	KOLLEKCIO	M, O
A művelet sikeressége	OK	Logikai	O
Segédváltozók	I, J	Egész	M

```
GENERAL (N, A, OK)
/* A mátrix létrehozása */
I<-0
while (I<N) AND VANHELY (MERET (SOR))
    I<-I+1
    HELYFOGLAL (A[I])
OK<-I=N
if OK
    /* A mátrix feltöltése */
    for I<-1, N
        for J<-1, N
            (*A[I]) [J]<-RANDOM (2)
else
    /* A lefoglalt memória felszabadítása */
    for J<-1, I
        FELSZABADIT (A[J])
```

3. MEGOLDANDÓ FELADATOK

Készítsünk teljesen paraméterezett szubrutint (adatszerkezeti táblázat és algoritmus) az alábbi feladatok megoldására!

- Egy nagy pozitív egész számot egy egyirányban láncolt lista tárolja úgy, hogy minden láncelem egy db számjegyet tárol.
 - Egy adott nagy szám osztható-e 4-gyel?
 - Osszuk el egy adott nagy számot (a szám megváltoztatása nélkül) 100-zal úgy, hogy két eredmény keletkezzen, a hányados és maradék!
- Töröljük ki egy $N \times M$ -es ($N \leq 500$, $M \leq 500$) „nagy” mátrix adott indexű sorát!