Facultatea de Matematică şi Informatică
Algoritmi şi Structuri de Date – Laborator
Anul I, semestrul I, an universitar 2014/2015
Grupe: 141
Web: http://laborator.wikispaces.com

Tema săptămânii 2
8 octombrie 2014

Probleme obligatorii

Termen de predare : Laboratorul din săptămâna 4 (24 octombrie 2014)

(3 p) **1.** Să se scrie un program care implementează liste simplu înlănțuite alocate dinamic. Vor exista funcții pentru următoarele operatii:

(a) adăugarea unui nou element;
(b) afişarea listei;
(c) căutarea unui element;
(d) ştergerea unui element;
(e) ştergerea întregii liste.

(1 p) **2.** Scrieți funcții pentru inserarea unui nod într-o listă alocată dinamic:
a) la început;
b) în interiorul listei;
c) la finalul listei.

Sugestii de implementare C++

Un nod al listei poate de următorul tip:

```
struct NOD {
        int value;
        NOD *next;
}
```

Declararea si alocarea de memorie:

```
NOD *unu, *doi;
unu = new NOD;
doi = new NOD;
```

Modifcarea câmpurilor nodului şi eliberarea memoriei se fac cu instrucțiunile:

```
unu->value = 1;
unu->next = doi;
```

```
doi->value = 2;
doi->next = NULL;

delete unu;
delete doi;
```

Probleme suplimentare

Termen de predare : Laboratorul din săptămâna 4 (24 octombrie 2014)

(1 p) **3.** Traversarea unei structuri liniare alocate static (cu prelucrare de chei)
Scrieți subprograme pentru:

a) găsirea elementului maxim din structură
b) găsirea poziției elementului minim din structură.

Algoritm

```
procedure Traversare(A, 1, n)
k := 1; {inițializarea indicelui pentru traversare}
while k <= n do {test pentru nedepăşirea structurii}
     vizitează A[k];
     k := k+1; {trecem la componenta următoare}
endwhile
endproc
```

(1 p) **4.** Inserarea unui element într-o structură liniară alocată static

```
procedure Insert(A, 1, n, k, Elem)
{inserează în structura liniară A[1 .. n], pe poziția k,
valoarea lui Elem}
     {mută pe rând elementele de la A[n] până la A[k]
câte o locație      la dreapta}
     i := n;
     while i >= k do
          A[i+1] := A[i];
          i := i-1;
     endwhile
     {inserarea propriu-zisă}
     A[k] := Elem;
     {creşte dimensiunea structurii}
     n := n+1;
endproc
```

(1 p) **5.** Ştergerea unui element dintr-o structură liniară alocată static

```
procedure Delete(A, 1, n, k, X)
{extrage în X valoarea A[k] şi reface vectorul}
```

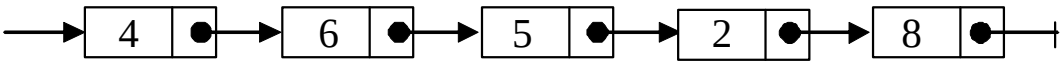
```
        {extragerea propriu-zisă}
        X := A[k];
        {refacerea structurii de vector}
        for i := k to n-1 do
             A[i] := A[i+1];
        endfor
        {scade dimensiunea structurii}
        n := n-1;
endproc
```

(2 p) **6.** Fiind dat un tablou ordonat cu *n* întregi în care o cheie poate să apară de mai multe ori, să se elimine cheile duble prin deplasări de elemente.

O reprezentare prin coeficienți a unui polinom $P = \sum_{k=0}^{n} a_k X^k$ de grad n este un **vector de coeficienti** $a = (a_0, a_1, ..., a_n)$. Fiind date polinoamele *P* si *Q* de grad n, respectiv m, reprezentate prin coeficienți, să se scrie un program care calculează :

(2 p) **7**. produsul lor;
(2 p) **8**. $P(x_0)$, adică evaluează polinomul *P* intr-un punct dat x_0.

(2 p) **9.** Reprezentarea numerelor mari (numere întregi cu număr mare de cifre) cu ajutorul unei liste liniare simplu înlănțuite se face folosind următoarea schemă:



Numărul întreg 82564 este reprezentat ca lista punînd fiecare cifră în câte un nod.

Scrieți un program în care se citesc două numere “mari” şi se construieşte o listă în care se va salva suma lor.

(3 p) **10**. Să se implementeze cu ajutorul unei **liste liniare simplu înlănțuită alocată dinamic** un polinom de grad n. Fiecare nod se va considera că reține gradul fiecărui monom, precum şi coeficientul său.

Structura poate fi definită astfel :

```
struct pol {
     int gr, coef;
     pol *next;
     };
```

Scrieți un program care creează două polinoame implementate prin liste şi calculează şi afişează suma lor.

Notă: Pentru algoritmi care rezolvă problemele 7 şi 8 folosind alocarea dinamică se acordă bonus 1 punct (pentru problema 7), respectiv 2 puncte (pentru problema 8).