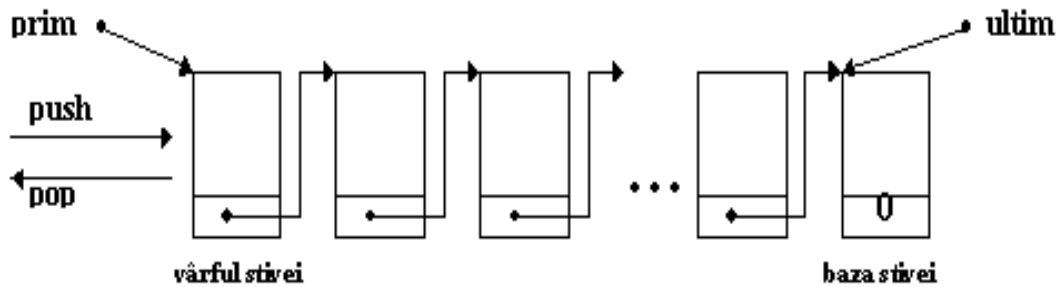


Laborator 3

Stive implementate dinamic

O **stivă** se definește ca o listă liniară simplu înlănțuită în care toate intrările și ieșirile se fac pe la un singur capăt al ei. Stiva este o structură de tip LIFO (Last In First Out), adică ultimul nod introdus este primul scos. Rezultă că, înregistrarea de pe nivelul k reține înregistrarea de pe nivelul k-1. În cazul stivei se reține doar elementul din vârful stivei.



Fiind o structură particulară a unei liste simplu înlănțuite, operațiile principale asupra unei stive sunt:

- **push** = adăugare - pune un element pe stivă; funcția se realizează prin inserarea unui nod înaintea primului;
- **pop** = eliminare - scoate elementul din vârful stivei; funcția se realizează prin ștergerea primului nod;
- **clear** - ștergerea stivei;

1. Sa se creeze o lista cu numere intregi folosind crearea prin adaugarea elementelor la inceputul listei (stiva). Se cere:

- a) Sa se afiseze continutul listei
- b) Sa se determine suma si maximul elementelor listei.

Exemplu:

Pentru n=5 si elementele {12, 10, -23, 100, -3}, se obtine suma = 96 si maximul = 100.

```
# include <iostream.h>
# include <conio.h>

struct lista
{
    int info;
    lista *leg;
};
```

```
lista *p, *prim, *ultim;
int n;

void creare(lista *&prim, lista *&ultim)
{
    int i,inf;
    cout<<"Dati numarul de elemente al listei n = ";
    cin>>n;
    cout<<"Dati prima informatie "; cin>>inf;
    prim=new lista;
    prim->info=inf;
    prim->leg=NULL;
    ultim=prim;
    for(i=2;i<=n;i++)
    {
        cout<<"Dati informatia "<<i<<" = ";
        cin>>inf;
        p=new lista;
        p->info=inf;
        p->leg=prim;
        prim=p;
    }
}

void afisare(lista *prim)
{
    p=prim;
    while(p!=NULL)
    {
        cout<<p->info<<" ";
        p=p->leg;
    }
}

int suma(lista *prim)
{
    int suma=0;
    p=prim;
    while(p!=NULL)
    {
        suma+=p->info;
        p=p->leg;
    }
    return suma;
}

int maxim(lista *prim)
{
    int max=-10000;
```

```
p=prim;
while(p!=NULL)
{
    if( p->info > max ) max=p->info;
    p=p->leg;
}
return max;
}

int main(void)
{
    creare(prim, ultim);
    afisare(prim);
    cout<<endl<<"Suma elementelor = "<<suma(prim)<<endl;
    cout<<endl<<"Maximul elementelor = "<<maxim(prim)<<endl;
}
```

2. Sa se creeze o lista cu numere intregi folosind crearea prin adaugare la sfarsitul listei. Se cere:
- a) Sa se afiseze continutul listei
 - b) Sa se determine numarul de numere prime

```
# include <iostream.h>

struct lista
{
    int info;
    lista *leg;
};

lista *p, *prim, *ultim;
int n;

void creare(lista *&prim, lista *&ultim)
{
    int i,inf;
    cout<<"Dati numarul de elemente al listei n = "; cin>>n;
    cout<<"Dati prima informatie "; cin>>inf;
    prim=new lista;
    prim->info=inf;
    prim->leg=NULL;
    ultim=prim;
    for(i=2;i<=n;i++)
    {
        cout<<"Dati informatia "<<i<<" = ";
        cin>>inf;
    }
}
```

```
    p=new lista;
    p->info=inf;
    p->leg=NULL;
    ultim->leg=p;
    ultim=p;
}
}
void afisare(lista *prim)
{
    p=prim;
    while(p!=NULL)
    {
        cout<<p->info<<" ";
        p=p->leg;
    }
}
int verific_prim(int x)
{
    int q=1;
    for(int i=2;i<=x/2;i++)
        if(x%i==0)    q=0;
    return q;
}
int nr_prime(lista *prim)
{
    int nr=0;
    p=prim;
    while(p!=NULL)
    {
        if(verific_prim(p->info)==1) nr++;
        p=p->leg;
    }
    return nr;
}
int main(void)
{
    creare(prim,ultim);
    afisare(prim);
    cout<<endl<<"numarul de numere prime = "<<nr_prime(prim)<<endl;
}
```

3. Sa se creeze o lista cu elemente litere mici. Se cere:
- a) afisarea listei
 - b) inlocuirea fiecarei vocale cu litera imediat urmatoare din alfabet si apoi sa se afiseze lista modificata
 - c) sa se transforme fiecare litera in litera mare si apoi sa se afiseze lista modificata.

```
# include <iostream.h>
struct lista
{
    char info;
    lista *leg;
};

lista *p, *prim, *ultim;
int n;

void creare(lista *&prim, lista *&ultim)
{
    int i;
    char inf;
    cout<<"Dati numarul de elemente al listei n = "; cin>>n;
    cout<<"Dati prima informatie "; cin>>inf;
    prim=new lista;
    prim->info=inf;
    prim->leg=NULL;
    ultim=prim;
    for(i=2;i<=n;i++)
    {
        cout<<"Dati informatia "<<i<<" = ";
        cin>>inf;
        p=new lista;
        p->info=inf;
        p->leg=NULL;
        ultim->leg=p;
        ultim=p;
    }
}

void afisare(lista *prim)
{
    p=prim;
    while(p!=NULL)
    {
        cout<<p->info<<" ";
        p=p->leg;
    }
}
```

```
void inlocuire(lista *prim)
{
    p=prim;
    char voc[5]="aeiou";

    while(p!=NULL)
    {
        for(int i=1;i<=5;i++)
            if(voc[i]==p->info)
                p->info=p->info+1;
        p=p->leg;
    }
    cout<<endl<<"lista dupa inlocuire: ";
    afisare(prim);
}

void transformare(lista *prim)
{
    p=prim;
    while(p!=NULL)
    {
        p->info=p->info-32;
        p=p->leg;
    }
    cout<<endl<<"lista dupa transformare : ";
    afisare(prim);
}

int main(void)
{
    creare(prim,ultim);
    cout<<endl<<"lista initiala : ";
    afisare(prim);
    inlocuire(prim);
    transformare(prim);
}
```

4. Sa se creeze o lista cu primii nr termeni din sirul lui Fibonacci. Se cere:
Exemplu: Pentru n=7 se obtine lista cu elementele 1, 1, 2, 3, 5, 8, 13.

```
# include <iostream.h>
struct lista{
    int info;
    lista *leg;
};

lista *p, *prim, *ultim;
int nr;

void creare(lista *&prim, lista *&ultim, int nr){
    int i,j,k,inf;
    prim=new lista;
    prim->info=1;
    prim->leg=NULL;
    p=new lista;
    p->info=1;
    p->leg=NULL;
    prim->leg=p;
    ultim=p;
    i=2;j=1;k=1;
    while(i<=nr) {
        i=j+k;
        p=new lista;
        p->info=i;
        p->leg=NULL;
        ultim->leg=p;
        ultim=p;
        j=k;
        k=i;
    }
}

void afisare(lista *prim){
    p=prim;
    while(p!=NULL){
        cout<<p->info<<" ";
        p=p->leg;
    }
}

int main(void){
    cout<<"Dati numarul pana la care se calculeaza termenii sirului lui Fibonacci "; cin>>nr;
    creare(prim, ultim, nr);
    afisare(prim);
}
```

Probleme propuse spre rezolvare

1. Să se ruleze programele prezentate mai sus, urmărind apelurile și valorile parametrilor de apel.
2. Sa se implementeze un programa care sa exemplifice operatiile care se efectueaza intr-o coada, construind un program C++ cu meniuri specifice operatiile cu coada.
3. Se da o lista cu numere naturale. Sa se creeze doua liste, una cu numerele pare din lista initiala, iar cea de a doua cu numerele impare. Sa afiseze la final cele doua liste obtinute.
4. Sa se creeze o listă simplu înlănțuită cu numerele natuale preluate de la tastatura. Numerele sunt dispuse pe mai multe linii ale fișierului de intrare, cele de pe aceeași linie fiind despărțite de câte un spațiu. Se cere să se adauge după fiecare număr din listă, răsturnatul său. Să se afișeze conținutul listei înainte și după modificare, fiecare afișare pe câte un rând.
5. Să se creeze o listă simplu înlănțuită cu numerele întregi preluate de la tastatura. Numerele sunt dispuse pe mai multe linii ale fișierului de intrare, cele de pe aceeași linie fiind despărțite de câte un spațiu. Se cere să se adauge după fiecare număr din listă cifra de control a numărului respectiv. Să se afișeze conținutul listei înainte și după modificare, fiecare afișare pe câte un rând.
6. Eliminarea elementelor prime dintr-o lista.
7. Sa se creeze o listă liniară simplu înlănțuită care prelucreaza n numere intregi.
 - a) Sa se afiseze continutul listei
 - b) Sa se mai adauge 2 elemente in lista si sa se afiseze
 - c) Sa se mai adauge m elemente in lista si sa se afiseze
 - d) Sa se numere cate numere prime sunt in lista
 - e) Sa se determine suma (produsul) elementelor din lista
 - f) Sa se determine daca lista este ordonata crescator
 - g) Sa se determine minimul (maximul) listei
 - h) Sa se mai creeze o lista care se va adauga la sfarsitul listei (concatenare). Sa se afiseze lista noua astfel generate
 - i) Se vor dubla elementele pentru care suma cifrelor este mai mica decat 10
8. Sa se formeze o lista care contine elementele unei matrici.
9. Sa se formeze o lista care contine elementele unui vector
10. Sa se genereze o lista care va retine numere intregi si dupa fiecare numar suma cifrelor sale
11. O lista contine titlul si autorul unei carti, precum si editura. Sa se afiseze acele carti din lista editate la editura Academica Brancusi din Targu-Jiu.

PROIECTAREA ALGORITMILOR

Laborator3

12. Data fiind o lista cu elemente numere intregi, sa se formeze o a doua lista care contine elementele distincte din lista data, impreuna cu frecventa lor de aparitie.

13. Sa se creeze o lista care prelucreaza numere complexe. Lista se citeste din fisier pe fiecare linie fiind scrise doua valori reprezentand partea reala si respective cea zecimala

- a) Sa se afiseze continutul listei
- b) Sa se determine numarul complex cu modulul cel mai mare
- c) Sa se determine suma (produsul) elementelor din lista

14. Fie o lista de numere intregi distincte. Primele doua valori (cele mai mari) se vor insera la inceputul listei.

Exemplu:

Daca lista retine initial 1, 3, 5, 4, 9, 25, 14, 32, 11 atunci se va obtine: 25, 32, 1, 3, 5, 4, 9, 25, 14, 32, 11

15. Sa se afiseze elementele (elementul) din mijlocul unei liste de intregi

16. In doua fisiere text F1.IN si F2.IN se gasesc 2 multimi A si respectiv B cu elemente numere naturale.

- a) Sa se creeze cate o lista liniara simplu inlantuita pentru memorarea elementelor din fiecare dintre cele 2 multimi.
- b) Sa se creeze o lista liniara simplu inlantuita care se memoreze reuniunea celor 2 multimi.
- c) Sa se creeze o lista liniara simplu inlantuita care se memoreze intersectia celor 2 multimi.