Structura de tip Stiva(Stack)

Aceasta structura de date este un caz particular de lista care functioneaza pe principiul LIFO (last in – first out, ultimul intrat este primul servit, puteti sa va ganditi la o stiva de materiale pentru care un nou material se va adaugaintotdeauna deasupra si se va extrage tot de deasupra).

Prin urmare principalele prelucrari care se refera la aceasta structura de date vor fi:

- creare stiva
- listare stiva (parcurgere in ordine inversa creerii)
- adaugare la sfarsit (peste varful stivei, operatie numita ***push ( )***)
- stergere element din varful stivei (operatie numita ***pop( )*** **)**
- prelucrarea varfului stivei

Specific acestei structuri de date este faptul ca prelucrarile se fac intotdeauna la elementul de la acelasicapat, element pe care il vom numi varf.

Functia *push( )* , creaza stiva cand aceasta este vida sau adauga un nou element in caz contrar.

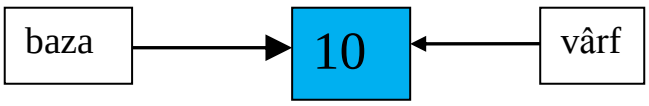
Functia *pop( )*, elimina elementul din varful stivei.

Fie o stiva de numere intregi pentru care elementele sunt adaugate in ordinea: 10,20,30,40

La primul pas se creaza stiva caz in care primul element creat va fi varful stivei:

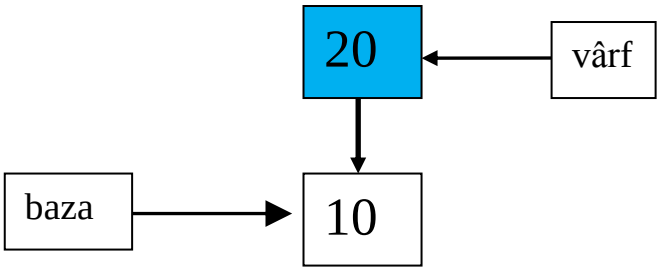
Se introduce primul element in stiva(10)

Acum stiva este:

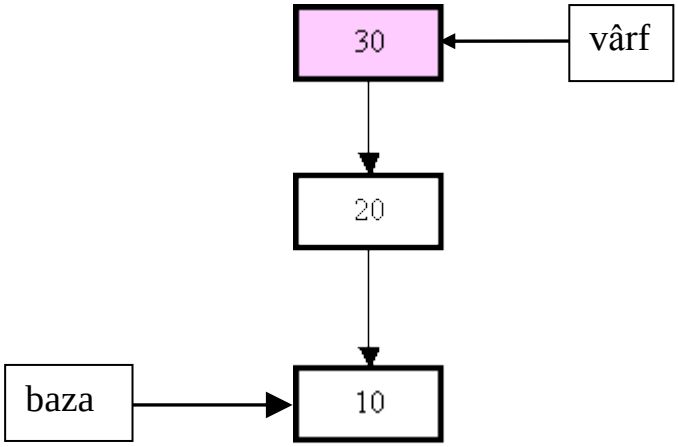


Apoi se adauga prin operatia***push ( )*** un nou element, 20:

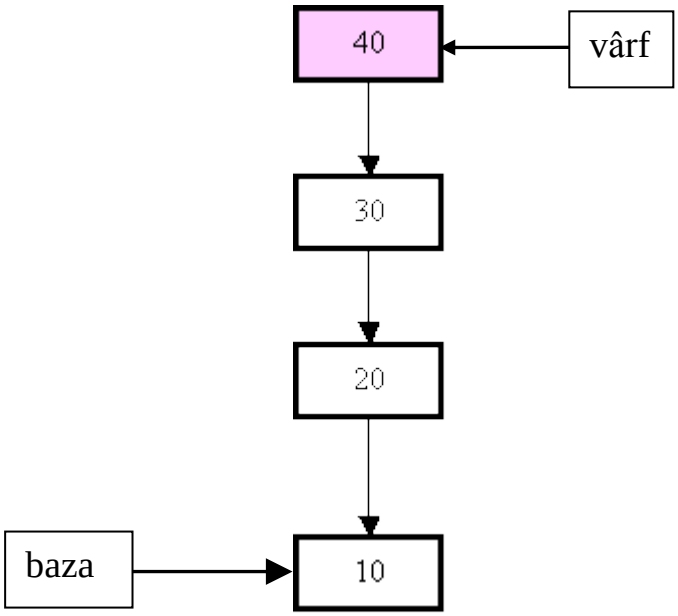
a) Se creste nivelul varfului
b) Se introduce valoarea in stiva



In continuare se adaugaprin operatia***push ( )*** un nou element, 30 care va deveni noul varf:



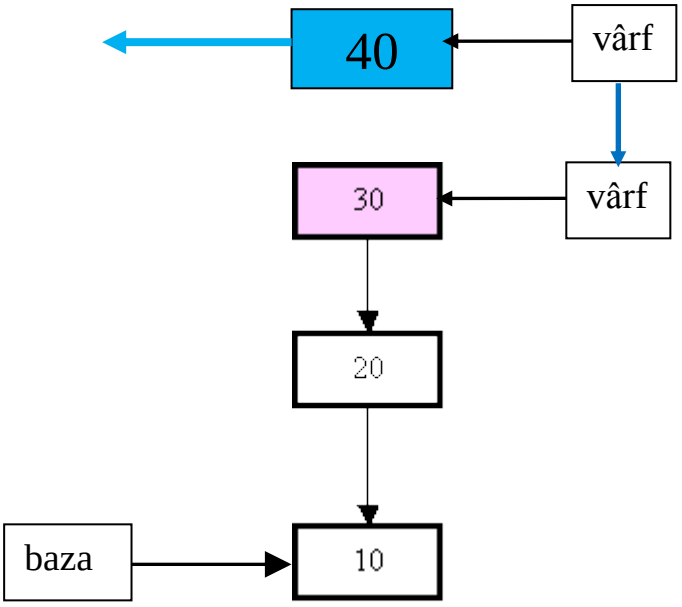
La sfarsit se adaugaprin operatia***push ( )*** un nou element, 40şi stiva devine:



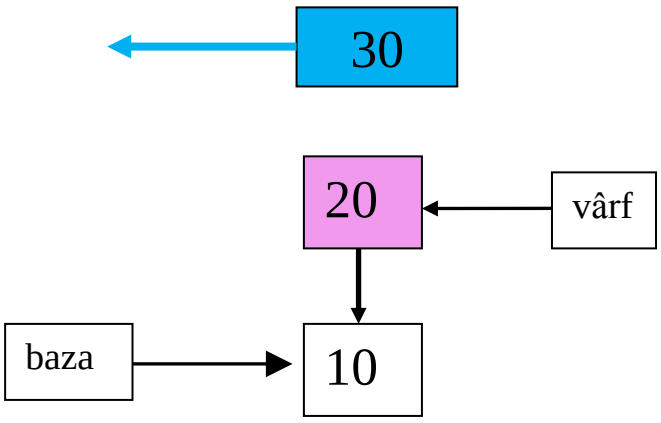
Afisarea se va face in ordine inversa generarii si prin urmare se va afisa: 40,30,20, 10.

Pentru stergere, prin operatia**pop( )** se va sterge elementul din varful stivei, 40,dupa care noul varf va deveni precedentul sau, 30:
Funcţia pop presupune :

a. Se reţine adresa vârfului stivei
b. Se coboară pe stivă modificând vârful stivei ca fiind nodul de sub vârf
c. Se şterge fostul vârf al stivei



Un nou apel al functiei pop va elimina elementul din vârful stivei.



Pentru a prelucra aceasta structura de date va fi suficient un singur pointer, pentru varf.

O implementare statica

```
#include <iostream>

using namespace std;

int stiva[100],varf;

void push()
{
   varf++;
   cout<<"Introdu valoarea: ";
   cin>>stiva[varf];
}

void pop()
{

   varf--;
}


void top()//prelucarea primului element din stiva
{
   cout<<stiva[varf]<<" ";
}

int main()
{
   int n,i;
   cout << "Numar de elemente in stiva= " ;
   cin>>n;
   for(i=1;i<=n;i++)
      push();
   cout<<"Elementele din stiva= ";
   while(varf!=0)
   {
      top();
      pop();
   }

   return 0;
}
```

O implementare dinamică:

```
//prelucrari stiva : (LIFO)

#include<iostream >

using namespace std;


struct nod
{
   int info;
   nod *ant;
};

nod *varf;

void push(nod* &v, int x)
{
   nod *q;
   if(!v)
   {
      v=new nod;
      v->info=x;
      v->ant=NULL;
   }
   else
   {
      q=new nod;
      q->ant=v;
      q->info=x;
      v=q;
   }
}

void afisare(nod *v)
{
   nod *q;
   q=v;
   while(q)
   {
      cout<<q->info<<" ";
      q=q->ant;
   }
```

```
}

void pop(nod* &v)
{
   nod* q;
   if(!v)
      cout<<"stiva este vida si nu mai ai ce elimina!!!";
   else
   {
      q=v;
      v=v->ant;
      delete q;
   }
}


int main()
{
   int n,a,i;
   cout<<"numarul initial de noduri ";
   cin>>n;
   for(int i=1;i<=n;i++)
   {
      cout<<"valoarea de adaugat in stiva ";
      cin>>a;
      push(varf,a);
   }
   cout<<endl<<"Stiva este: ";
   afisare(varf);
   int nre, nra;
   cout<<endl<<"cate adaugari ? ";
   cin>>nra;
   for(i=1;i<=nra; i++)
   {
      cout<<"valoarea de adaugat ";
      cin>>a;
      push(varf,a);
   }
   cout<<endl<<"dupa adaugare "<<endl;
   n=n+nra;
   cout<<"stiva are  "<<n<<" elemente"<<endl;
   cout<<endl<<"Stiva este: ";
   afisare(varf);
   cout<<endl<<"cate eliminari ?";
   cin>>nre;
```

```
    for(i=1;i<=nre;i++)
       pop(varf);
    cout<<endl<<"dupa eliminare"<<endl;
    n=n-nre;
    cout<<"stiva are  "<<n<<" elemente"<<endl;
    cout<<endl<<"Stiva este: ";
    afisare(varf);
    //prelucrez varful stivei: de exemplu se poate dubla conținutul:
    varf->info=2*varf->info;
    cout<<endl<<"Dupa dublarea valorii varfului "<<"stiva este: ";
    afisare(varf);
    return 0;
}
```

Probleme propuse:

1. Se citesc siruri de caractere pana la “*”. Sa se afisezesirurile in ordine inversa citirii. Sa concateneze “blabla” la ultimul sir citit. Sa se stearga ultimele 3 siruri citite apoi sa se afiseze

2.Sa se memoreze n numere intregiintr-o structura de tip stiva . Sa se stearga elementele din varfulstivei pana se intalneste un numar pentru care suma cifrelor este mai mare decat 10

3. Sa se prelucreze datele celor n angajati ai unei firme (nume, profesia, salariu). Datele se citesc din fisier si sunt inregistrate in ordinea angajarii (pentru fiecare persoana datele sunt scrise pe o linie).

a) Sa se afiseze datele dupa ce acestea au fost memorate intr-o structura de tip stiva.
b) Sa se afiseze ultimii 3 angajati
c) Cel mai recent angajat este concediat. Sa se afiseze si sa modifice datele din structura si apoi din fisier