Enostransko povezan seznam (single linked list)

Deklaracija podatkovnega tipa elementa seznama :

**struct element**

**{ tip1 podatek1;**

**tip2 podatek2;**

**.......**

**tipN podatekN;**

**struct element \*nasl; } \*kazalec;**

start

NULL

**11**

….

**6**

**4**

**vstopna točka podatek kazalčna**

**seznama komponenta**

### Neurejeno povezan enostranski seznam

struct element

{

int data;

struct element \*nasl;

}\*Start=NULL;

1. dodajanje elementa v seznam (vedno na začetek) **(**funkcija **push)**

Dinamična rezervacija za novi element seznama.

Kazalčni komponenti novega elementa priredimo vrednost kazalčne spremenljivke start.

Kazalčni spremenljivki start priredimo vrednost kazalčne spremenljivke tmp.

void Push (int a)

{

//tmp=(struct element \*) malloc (sizeof(struct element));

tmp=new struct element;

tmp->data=a;

if (Start==NULL) //preverjanje, če je seznam prazen

{

Start=tmp;

tmp->nasl=NULL;

}

else //če seznam vsebuje vsaj en element

{

tmp->nasl=Start;

Start=tmp;

}

}

//lahko tudi na krajši način:

void Push (int a)

{

//tmp=(struct element \*) malloc (sizeof(struct element));

tmp=new struct element;

tmp->data=a;

tmp->nasl=Start;

Start=tmp;

}

1. brisanje iz seznama **(**brišemo začetni element) **(**funkcija **pop)**

void Pop ()

{

if (Start != NULL)

{

tmp=Start;

Start=Start->nasl;

delete tmp;

//free(tmp);

}

}

1. Izpis seznama: //p je globalen in funkcijo kličemo znotraj zanke v main()

void Izpis ()

{

if(p!= NULL)

{

cout<< p->data <<" ";

p=p->nasl;

}

}

Izpis() je lahko tudi drugačen (seveda je treba ustrezno spremeniti main()):

void Izpis ()

{

p=Start;

while(p!= NULL)

{

cout<< p->data <<" ";

p=p->nasl;

}

}

Primer:

#include <iostream>

using namespace std;

struct element

{

int data;

struct element \*nasl;

} \*p,\*tmp,\*Start;

void Push (int a)

{

tmp = new struct element;

tmp->data=a;

tmp->nasl=Start;

Start=tmp;

}

void Izpis ()

{

cout<< p->data <<" ";

p=p->nasl;

}

int main ()

{

int i;

Start=(struct element \*)NULL; //Start=NULL;

do

{

cin >>i;

if (i==0)

break;

Push(i);

}

while (i!=0);

p=Start;

cout<<endl;

while(p!=NULL)

Izpis();

return 0;

}