MINISTRY OF EDUCATION OF THE REPUBLIC OF BELARUS

EDUCATIONAL INSTITUTION

«BREST STATE TECHNICAL UNIVERSITY»

Department of IIT

**Laboratory work №14**

**For the second semester**

**Topic: «Memory Allocation»**

Completed by the 1st year student of

Faculty of Electronic Information Systems

the group AC-57f Chernookiy I.V.

Checked by Khatskevich M.V.

Brest 2019

**Laboratory work №14**

**Topic: «Memory Allocation»**

**Goal:** To learn the main properties of the memory allocation and try to use them in practice.

**Task 1.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct node

{

int info;

node \*next;

} \*start, \*newptr, \*save, \*ptr;

node \*create\_new\_node(int);

void insert\_at\_beg(node \*);

void display(node \*);

void main()

{

clrscr();

start = NULL;

int inf;

char ch='y';

while(ch=='y' || ch=='Y')

{

clrscr();

cout<<"Enter Information for the new node: ";

cin>>inf;

cout<<"\nCreating new node..!!..Press any key to continue..";

getch();

newptr = create\_new\_node(inf);

if(newptr != NULL)

{

cout<<"\n\nNew node created successfully..!!\n";

cout<<"Press any key to continue..";

getch();

}

else

{

cout<<"\nSorry..!!..cannot create new node..!!..Aborting..!!";

cout<<"\nPress any key to exit..";

getch();

exit(1);

}

cout<<"\n\nNow inserting this node at the beginning of the list..\n";

cout<<"Press any key to continue..\n";

getch();

insert\_at\_beg(newptr);

cout<<"\nNode successfully inserted at the beginning of the list.\n";

cout<<"Now the list is:\n";

display(start);

cout<<"\nWant to enter more nodes ? (y/n)..";

cin>>ch;

}

getch();

}

node \*create\_new\_node(int n)

{

ptr = new node;

ptr->info = n;

ptr->next = NULL;

return ptr;

}

void insert\_at\_beg(node \*np)

{

if(start==NULL)

{

start = np;

}

else

{

save = start;

start = np;

np->next = save;

}

}

void display(node \*np)

{

while(np != NULL)

{

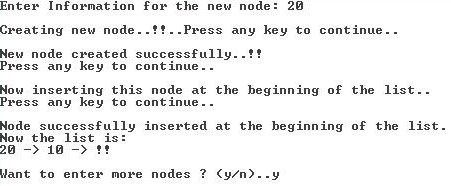
cout<<np->info<<" -> ";

np = np->next;

}

cout<<"!!\n";

}



**Task 2.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct node

{

int info;

node \*next;

} \*start, \*newptr, \*save, \*ptr, \*rear;

node \*create\_new\_node(int);

void insert\_node(node \*);

void display\_node(node \*);

void delete\_node();

void main()

{

clrscr();

start = rear = NULL;

int inf;

char ch='y';

while(ch=='y' || ch=='Y')

{

clrscr();

cout<<"Enter Information for the new node: ";

cin>>inf;

newptr = create\_new\_node(inf);

if(newptr == NULL)

{

cout<<"\nSorry..!!..cannot create new node..!!..Aborting..!!";

cout<<"\nPress any key to exit..";

getch();

exit(1);

}

insert\_node(newptr);

cout<<"\nWant to enter more nodes ? (y/n)..";

cin>>ch;

}

clrscr();

do

{

cout<<"The list now is:\n";

display\_node(start);

cout<<"\nWant to delete first node ? (y/n)..";

cin>>ch;

if(ch=='y' || ch=='Y');

{

delete\_node();

}

}while(ch=='y' || ch=='Y');

getch();

}

node \*create\_new\_node(int n)

{

ptr = new node;

ptr->info = n;

ptr->next = NULL;

return ptr;

}

void insert\_node(node \*np)

{

if(start==NULL)

{

start = rear = np;

}

else

{

rear -> next = np;

rear = np;

}

}

void delete\_node()

{

if(start == NULL)

{

cout<<"Underflow...!!\n";

}

else

{

ptr = start;

start = start->next;

delete ptr;

}

}

void display\_node(node \*np)

{

while(np != NULL)

{

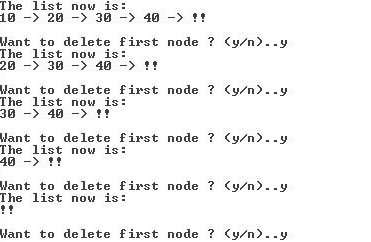
cout<<np->info<<" -> ";

np = np->next;

}

cout<<"!!\n";

}



**Task 3.**

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct node

{

int info;

node \*next;

} \*start, \*newptr, \*save, \*ptr, \*rear;

node \*create\_new\_node(int);

void insert\_node(node \*);

void travers(node \*);

void main()

{

clrscr();

start = rear = NULL;

int inf;

char ch='y';

while(ch=='y' || ch=='Y')

{

cout<<"Enter Information for the new node: ";

cin>>inf;

newptr = create\_new\_node(inf);

if(newptr == NULL)

{

cout<<"\nSorry..!!..cannot create new node..!!..Aborting..!!";

cout<<"\nPress any key to exit..";

getch();

exit(1);

}

insert\_node(newptr);

cout<<"Want to enter more nodes ? (y/n)..";

cin>>ch;

cout<<"\n";

}

cout<<"The list now is:\n";

travers(start);

getch();

}

node \*create\_new\_node(int n)

{

ptr = new node;

ptr->info = n;

ptr->next = NULL;

return ptr;

}

void insert\_node(node \*np)

{

if(start==NULL)

{

start = rear = np;

}

else

{

rear -> next = np;

rear = np;

}

}

void travers(node \*np)

{

while(np != NULL)

{

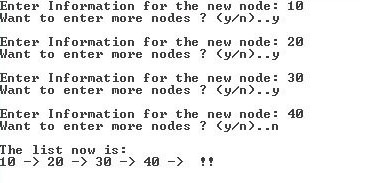
cout<<np->info<<" -> ";

np = np->next;

}

cout<<" !!\n";

}

****

**Task 4.**

#include<iostream.h>

#include<stdlib.h>

#include<conio.h>

int push(int [], int &, int);

void display(int [], int);

const int SIZE = 50;

void main()

{

clrscr();

int stack[SIZE], item, top=-1, res;

char ch='y';

while(ch=='y' || ch=='Y')

{

cout<<"Enter item for insertion: ";

cin>>item;

res = push(stack, top, item);

if(res == -1)

{

cout<<"Overflow..!!..Aborting..Press a key to exit..\n";

getch();

exit(1);

}

cout<<"Element inserted successfully..!!\n";

cout<<"\nThe Stack now is:\n";

display(stack, top);

cout<<"\nWant to enter more ? (y/n).. ";

cin>>ch;

}

getch();

}

int push(int stack[], int &top, int elem)

{

if(top == SIZE-1)

{

return -1;

}

else

{

top++;

stack[top] = elem;

}

return 0;

}

void display(int stack[], int top)

{

cout<<stack[top]<<" <-- "<<"\n";

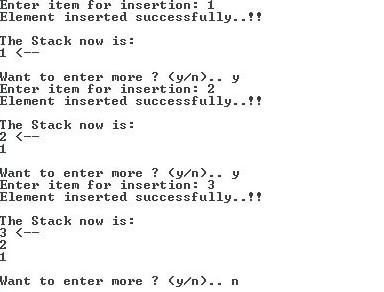
for(int i=top-1; i>=0; i--)

{

cout<<stack[i]<<"\n";

}

}

****