**STACKS, QUEUES AND LINKED LISTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **PROGRAMS** | **TEACHER’S**  **SIGNATURE** |
| 1. | Write a program to push and pop elements in a stack implemented as array. |  |
| 2. | Write a program to push and pop elements in a stack implemented as linked list. |  |
| 3. | Write a program to insert and delete elements in a queue implemented as array. |  |
| 4. | Write a program to insert and delete elements in a queue implemented as linked list. |  |

**STACKS, QUEUES AND LINKED LISTS**

1. Write a program to push and pop elements in a stack implemented as array.

## INPUT:

#include<iostream.h>

#include<conio.h>

void push(int stack[10],int &top,int item);

void pop(int stack[10],int &top);

void display(int stack[10],int top);

int top=-1;

int stack[100] ;

int size=10;

int main() {

clrscr();

int ele;

int no;

char op='y';

while(op=='y') {

cout <<"1. push 2.pop 3.display "<<endl;

cin>>no;

if (no==1){ cout <<"enter a number to push "<<endl; cin>>ele;

push(stack,top,ele); display(stack,top);

}

else if(no==2) { pop(stack,top);}

else { display(stack,top); }

cout <<"do you want to continue (y or n)"<<endl;

cin>>op;

}

getch();

}

void push(int stack[10],int &top,int item) {

if(top==size-1) {cout <<"overflow"<<endl; }

else {

top++;

stack[top]=item;

}

}

void pop(int stack[10],int &top) {

if(top==-1) {cout<<"underflow ";}

else { top-- ;}

}

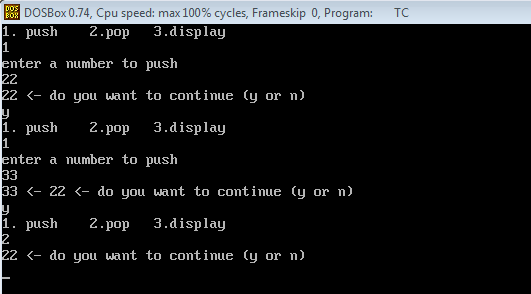
void display(int stack[10],int top) {

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

cout<<stack[i]<<" <- "; }

}

## OUTPUT:



1. Write a program to push and pop elements in a stack implemented as linked list.

## INPUT:

#include<iostream.h>

#include<conio.h>

#include<process.h>

struct node{

int info;

node \*next;

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

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

void push(node\*);

void pop();

void display(node\*);

void main()

{

clrscr();

int inf;

char ch='y';int choice;

top=NULL;

while(ch=='y')

{ cout<<"1.push 2.pop 3.display "<<endl; cin>>choice;

if(choice==1){

cout<<"Enter the information for node"<<endl; cin>>inf;

newptr=create\_new\_node(inf);

if(newptr==NULL)

cout<<"Cannot create new node"<<endl;

push(newptr); }

if(choice==2) {pop();display(top);}

else{

display(top);}

cout<<" do you want to continue (y/n) "<<endl;cin>>ch;

}

getch();

}

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

{

ptr=new node;

ptr->info=n;

ptr->next=NULL;

return ptr;

}

void push(node \*np)

{

if(top==NULL)

top=np;

else

{

save=top;

top=np;

np->next=save;

}

}

void display(node \*np)

{

cout<<" stack is "<<endl;

while(np!=NULL)

{

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

np=np->next;

}

cout<<"!!!!"<<endl;

}

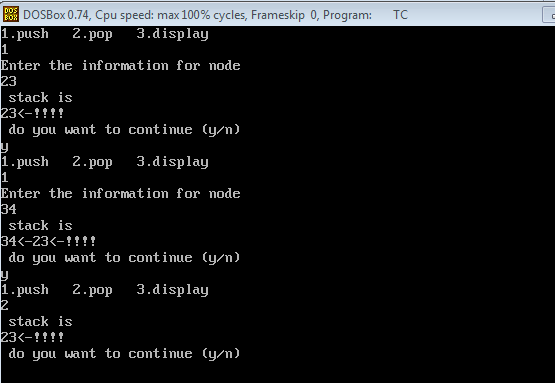
void pop() {

if(top==NULL){cout<<"underflow"<<endl;}

else { ptr=top; top=top->next; delete ptr; }

}

## OUTPUT:



1. Write a program to insert and delete elements in a queue implemented as array.

## INPUT:

#include <iostream.h>

#include <conio.h>

#include <process.h>

int remove(int[]);

int insert(int[],int);

void display(int[],int,int);

const int size=50;

int queue[size],front=-1,rear=-1;

int main() {

clrscr();

int item,res;int choice;

char ch='y';

while(ch=='y'){

cout <<"1. insert 2.delete 3.display "<<endl; cin>>choice;

if(choice==1){ cout<<"enter an item "<<endl; cin>>item;

res=insert(queue,item);

if(res==-1){cout<<"overflow....aborting ";exit(0); }

display(queue,front,rear);

}

else if(choice==2){

res=remove(queue);

if(res==-1){cout<<"underflow...aborting";exit(0);}

else{cout<<"element deleted is "<<res<<endl;

display(queue,front,rear);

}}

else if(choice==3){display(queue,front,rear); }

else {cout<<"error"<<endl;}

cout<<" do you want to continue (y/n) "<<endl; cin>>ch;

}

getch();

}

int insert(int queue[],int ele) {

if(rear==size-1){return -1; }

else if(rear==-1) {

front=rear=0;

queue[rear]=ele;

}

else{ rear++;

queue[rear]=ele; }

return 0;

}

int remove(int queue[]) {

int ret;

if(front==-1){return -1;}

else {

ret=queue[front];

if(front==rear){front=rear=-1;}

else{front++;}}

return ret;}

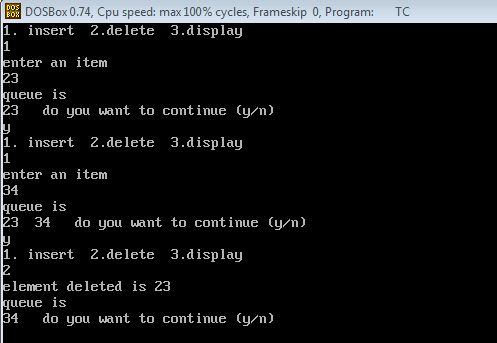
void display(int queue[],int front,int rear){

cout<<"queue is "<<endl;

for(int i=front;i<=rear;i++){cout<<queue[i]<<" ";}

}

## OUTPUT:



## Write a program to insert and delete elements in a queue implemented as linked list.

## INPUT:

#include <iostream.h>

#include <conio.h>

#include <process.h>

struct node {

int info;

node \*next;

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

node\* create(int a);

void insert(node \*);

void display(node \*);

void del();

int main(){

clrscr();

front=rear=NULL; int choice=0;

int inf;

char ch='y';

while(ch=='y') {

cout <<"1.insert 2.delete 3.display "<<endl; cin>>choice;

if(choice==1){

cout <<"enter information of new node "<<endl;cin>>inf;

newptr=create(inf);

if(newptr==NULL){ cout<<" aborting...error "<<endl; exit(0); }

insert(newptr);

}

else if(choice==2){ del();

display(front);

}

else if(choice==3){display(front);}

else{cout<<"incorrect choice "; exit(0); }

cout<<"do you want to continue (y/n) "<<endl; cin>>ch;}

getch();

}

node \*create(int a) {

ptr= new node;

ptr->info=a; ptr->next=NULL;

return ptr;

}

void insert(node \*np){

if(front==NULL){front=rear=np;}

else{rear->next=np; rear=np; }

}

void del(){

if(front==NULL){cout<<"underflow "<<endl;}

else{

ptr=front;

front=front->next;

delete ptr;

}}

void display(node \*np){

cout <<"queue is "<<endl;

while(np!=NULL) {cout<<np->info<<" ";

np=np->next;}

cout<<endl;

}

## OUTPUT:

