**Linear Stack**

#include <iostream.h>

#include<conio.h>

int stack[100], n=100, top=-1;

void push()

{

if(top>=n-1)

cout<<"Stack Overflow"<<endl;

else

{

cout<<"Enter value to be pushed:"<<endl;

cin>>val;

top++;

stack[top]=val;

}

}

void pop() {

if(top<=-1)

cout<<"Stack Underflow"<<endl;

else {

cout<<"The popped element is "<< stack[top] <<endl;

top--;

}

}

void display() {

if(top>=0) {

cout<<"Stack elements are:";

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

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

cout<<endl;

} else

cout<<"Stack is empty";

}

void main()

{

int ch, val;

cout<<"1) Push in stack"<<endl;

cout<<"2) Pop from stack"<<endl;

cout<<"3) Display stack"<<endl;

cout<<"4) Exit"<<endl;

do {

cout<<"Enter choice: "<<endl;

cin>>ch;

switch(ch)

{

case 1:

push(val);

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

cout<<"Exit"<<endl;

break;

default:

cout<<"Invalid Choice"<<endl;

}

}while(ch!=4);

}

**Linear Queue**

#include <iostream.h>

#include<conio.h>

int queue[100], n = 100, front = - 1, rear = - 1;

void Insert()

{

int val;

if (rear == n - 1)

cout<<"Queue Overflow"<<endl;

else {

if (front == - 1)

front = 0;

cout<<"Insert the element in queue : "<<endl;

cin>>val;

rear++;

queue[rear] = val;

}

}

void Delete() {

if (front == - 1 || front > rear)

{

cout<<"Queue Underflow ";

return ;

}

else

{

cout<<"Element deleted from queue is : "<< queue[front] <<endl;

front++;;

}

}

void Display()

{

if (front == - 1)

cout<<"Queue is empty"<<endl;

else {

cout<<"Queue elements are : ";

for (int i = front; i <= rear; i++)

cout<<queue[i]<<" ";

cout<<endl;

}

}

void main()

{

int ch;

cout<<"1) Insert element to queue"<<endl;

cout<<"2) Delete element from queue"<<endl;

cout<<"3) Display all the elements of queue"<<endl;

cout<<"4) Exit"<<endl;

do {

cout<<"Enter your choice : "<<endl;

cin>>ch;

switch (ch) {

case 1: Insert();

break;

case 2: Delete();

break;

case 3: Display();

break;

case 4: cout<<"Exit"<<endl;

break;

default: cout<<"Invalid choice"<<endl;

}

} while(ch!=4);

}

**Double Ended Queue**

#include<iostream.h>

#include<conio.h>

#define SIZE 10

class dequeue

{

int a[20],f,r;

public:

dequeue();

void insert\_at\_beg(int);

void insert\_at\_end(int);

void delete\_fr\_front();

void delete\_fr\_rear();

void show();

};

dequeue::dequeue()

{

f=-1;

r=-1;

}

void dequeue::insert\_at\_end(int i)

{

if(r>=SIZE-1) {

cout<<"\ninsertion is not possible, overflow!!!!";

} else {

if(f==-1) {

f++;

r++;

} else {

r=r+1;

}

a[r]=i;

cout<<"\n Inserted item is"<<a[r];

}

}

void dequeue::insert\_at\_beg(int i)

{

if(f==-1) {

f=0;

a[++r]=i;

cout<<"\n inserted element is:"<<i;

}

else if(f!=0)

{

a[--f]=i;

cout<<"\n inserted element is:"<<i;

}

else

{

cout<<"\n insertion is not possible, overflow!!!";

}

}

void dequeue::delete\_fr\_front()

{

if(f==-1)

{

cout<<"\n deletion is not possible::dequeue is empty";

return;

}

else

{

cout<<"\n the deleted element is:"<<a[f];

if(f==r) {

f=r=-1;

return;

}

else

f=f+1;

}

}

void dequeue::delete\_fr\_rear()

{

if(f==-1)

{

cout<<"\n deletion is not possible::dequeue is empty";

return;

}

else

{

cout<<"\n the deleted element is:"<<a[r];

if(f==r)

{

f=r=-1;

}

else

r=r-1;

}

}

void dequeue::show()

{

if(f==-1)

{

cout<<"\n Dequeue is empty";

} else

{

for(int i=f;i<=r;i++)

{

cout<<a[i]<<" ";

}

}

}

void main()

{

int c,i; clrscr();

dequeue d;

cout<<" 1.insert at beginning\n";

cout<<" 2.insert at end\n";

cout<<" 3.show\n";

cout<<" 4.deletion from front\n";

cout<<" 5.deletion from rear\n";

cout<<" 6.exit";

while(c!=6)

{

cout<<"\n enter your choice:\n";

cin>>c;

switch(c)

{

case 1:

cout<<"enter the element to be inserted\n";

cin>>i;

d.insert\_at\_beg(i);

break;

case 2:

cout<<"enter the element to be inserted\n";

cin>>i;

d.insert\_at\_end(i);

break;

case 3:

d.show();

break;

case 4:

d.delete\_fr\_front();

break;

case 5:

d.delete\_fr\_rear();

break;

case 6:cout<<"exit";

break;

default:

cout<<"invalid choice";

break;

}

}

}

**Circular Queue**

#include <iostream.h>

#include<conio.h>

int cqueue[5];

int front = -1, rear = -1, n=5;

void insertCQ(int val)

{

if ((front == 0 && rear == n-1) || (front == rear+1))

{

cout<<"Queue Overflow";

return;

}

if (front == -1) {

front = 0;

rear = 0;

}

else

{

if (rear == n - 1)

rear = 0;

else

rear = rear + 1;

}

cqueue[rear] = val ;

}

void deleteCQ()

{

if (front == -1)

{

cout<<"Queue Underflow";

return ;

}

cout<<"Element deleted from queue is : "<<cqueue[front]<<endl;

if (front == rear)

{

front = -1;

rear = -1;

} else {

if (front == n - 1)

front = 0;

else

front = front + 1;

}

}

void displayCQ()

{

int f = front, r = rear;

if (front == -1)

{

cout<<"Queue is empty"<<endl;

return;

}

cout<<"queue elements are :";

if (f <= r) {

while (f <= r){

cout<<cqueue[f]<<" ";

f++;

}

}

else

{

while (f <= n - 1)

{

cout<<cqueue[f]<<" ";

f++;

}

f = 0;

while (f <= r)

{

cout<<cqueue[f]<<" ";

f++;

}

}

cout<<endl;

}

int main() {

int ch, val;

cout<<"1)Insert ";

cout<<"2)Delete";

cout<<"3)Display ";

cout<<"4)Exit ";

do {

cout<<"Enter choice : "<<endl;

cin>>ch;

switch(ch)

{

case 1:

cout<<"Input for insertion: "<<endl;

cin>>val;

insertCQ(val);

break;

case 2:

deleteCQ();

break;

case 3:

displayCQ();

break;

case 4:

cout<<"Exit ";

break;

default: cout<<"Incorrect! ";

}

}

while(ch != 4);

return 0;

}