**Assignment # 06**

**Name : Noman Amjad.**

**Class: BSCS-3(B).**

**Arid: 21-Arid-654.**

**Date: 20-12-2022.**

**Question 1:**

**Ristricted input queue And Ristricted output queue:**

Code:

#include<iostream>

using namespace std;

class queue

{

int n;

int A[5];

int front;

int rear;

public:

queue()

{

front=rear=-1;

n=5;

}

void Enqueue()

{

if(rear==n-1)

{

cout<<"\nQueue is Full...\n";

}

else

{

int v;

cout<<"\nEnter a value : ";

cin>>v;

if(rear==-1)

{

front=0;

rear++;

A[rear]=v;

}

else

{

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

{

A[i+1]=A[i];

}

A[0]=v;

rear++;

}

}

}

void display()

{

if(front==-1 && rear==-1)

{

cout<<"\nQueue is Empty....!!!\n";

}

cout<<"\n\*\*\*\*\*Queue Element\*\*\*\*\*\*\n";

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

{

cout<<A[i]<<endl;

}

}

void Dequeue()

{

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

{

cout<<"\nQueue is Empty..!!\n";

}

else

{

cout<<"\nElement "<<A[front]<<" has been De Queue() succussfully...\n";

for(int i=0;i<=rear;i++)

{

A[i]=A[i+1];

front=0;

}

}

}

};

int main()

{

queue q;

int choice;

do

{

cout<<"\n\*\*\*\*Restricted input And output\*\*\*\*\*\n";

cout<<"\nPress 1 to EnQueue: ";

cout<<"\nPress 2 to DeQueue: ";

cout<<"\nPress 3 to Display : ";

cout<<"\nEnter a Choice :";

cin>>choice;

switch(choice)

{

case 1:

q.Enqueue();

break;

case 2:

q.Dequeue();

break;

case 3:

q.display();

break;

default:

cout<<"\nInvalid Input!!!!\n";

}

}while(choice!=0);

return 0;

}

**Question No 2: (Priority Queue)**

**Code:**

#include<iostream>

using namespace std;

class priorty\_queue;

priorty\_queue \*start=NULL;

class priorty\_queue

{

int num;

priorty\_queue \*next;

priorty\_queue \*pre;

public:

void Enqueue()

{

priorty\_queue \*t=new priorty\_queue();

cout<<"\nEnter a Number : ";

cin>>t->num;

if(start==NULL)

{

start=t;

t->pre=NULL;

t->next=NULL;

}

else

{

priorty\_queue \*s=start;

while(s!=NULL)

{

if(t->num>=s->num)

{

if(s->next!=NULL)

{

s=s->next;

}

else

{

s->next=t;

t->pre=s;

t->next=NULL;

break;

}

}

else

{

if(s->pre==NULL)

{

s->pre=t;

t->next=s;

start=t;

t->pre=NULL;

break;

}

else

{

priorty\_queue \*temp=new priorty\_queue();

t->next=s;

temp->next=s->pre;

s->pre=t;

t->pre=temp->next;

s=temp->next;

s->next=t;

break;

/\* t->pre=s->pre;

t->next=s->pre->next;

s->pre=t;

s->pre->next=t;\*/

}

}

}

}

}

void DeQueue()

{

if(start==NULL)

{

cout<<"\nEmpty Queue!!!\n\a";

}

else

{

priorty\_queue \*d=start;

start=start->next;

delete d;

cout<<"\nNode deleted succussfully...\n";

}

}

void display()

{

if(start==NULL)

{

cout<<"\nEmpty Queue!!!\n";

}

else

{

priorty\_queue \*t=start;

while(t!=NULL)

{

cout<<"\nNumber is : "<<t->num<<endl;

t=t->next;

}

}

}

};

int main()

{

priorty\_queue q;

int choice;

do

{

cout<<"\n\t\t\t\*\*\*\*\*\*\*\*\*Priorty Queue\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\nPress 1 for EnQueue : ";

cout<<"\nPress 2 for DeQueue : ";

cout<<"\nPress 3 for display : ";

cout<<"\nPress 0 for exit : ";

cout<<"\nEnter a choice : ";

cin>>choice;

switch(choice)

{

case 1:

q.Enqueue();

break;

case 2:

q.DeQueue();

break;

case 3:

q.display();

break;

default:

cout<<"\nInvalid input...!!!\n";

}

}while(choice!=0);

return 0;

}

**Question 3: (circular queue)**

**Code:**

#include<iostream>

using namespace std;

class Queue

{

int arr[5];

int rear;

int front;

public:

Queue()

{

front=-1;

rear=-1;

}

void enQueue()

{

if(isFull())

{

cout<<"\nQueue is overflow ..!!";

}

else

{

int val;

cout<<"\nEnter a Number : ";

cin>>val;

if(front=-1 && rear==-1)

{

front=0;

rear=0;

arr[rear]=val;

}

else

{

int r=rear;

while(arr[r]>val)

{

arr[r+1]=arr[r];

r=(r-1)%5;

}

arr[r+1]=val;

rear=(rear+1)%5;

}

}

}

void deQueue()

{

if(isEmpty())

{

cout<<"\nQueue is Underflow...!!!";

}

else

{

cout<<"\nThe Element "<<arr[front]<<" DeQueue succussfullay ....\n";

if(front==rear)

{

front=-1;

rear=-1;

}

else

{

front=(front+1)%5;

}

}

}

bool isFull()

{

if((rear+1)%5==front)

{

return 1;

}

else

{

return 0;

}

}

bool isEmpty()

{

if(front==-1 && rear==-1)

{

return true;

}

else

return false;

}

void display()

{

cout<<"\nThe Queue Element is : \n";

for(;front!=rear;front=(front+1)%5)

{

cout<<arr[front]<<" ";

}

cout<<arr[front];

}

};

int main()

{

Queue q;

int choice;

do

{

cout<<"\nPress 1 for Enqueue : ";

cout<<"\nPress 2 for Dequeue : ";

cout<<"\nPress 3 for display : ";

cout<<"\nEnter a choice : ";

cin>>choice;

switch(choice)

{

case 1:

q.enQueue();

break;

case 2:

q.deQueue();

break;

case 3:

q.display();

break;

default:

cout<<"\nInvalid Input!!!";

}

}while(choice!=0);

return 0;

}

**Question 4: (Simple Queue)**

**Code:**

#include<iostream>

using namespace std;

class queue

{

int que[5];

int n;

int front;

int rear;

public:

queue()

{

front=-1;

rear=-1;

n=5;

}

void insert()

{

int v;

if(rear==n-1)

{

cout<<"\nQueue is Overflow\n";

}

else

{

if(front==-1)

{

front=0;

}

cout<<"\nEnter a Value please : ";

cin>>v;

rear++;

que[rear]=v;

}

}

void deQueue()

{

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

{

cout<<"\nQueue is Underflow\n";

}

else

{

cout<<"\nElement "<<que[front]<<" Deleted...!!"<<endl;

front++;

}

}

void display()

{

if(front==-1)

{

cout<<"\nQueue IS Empty...\n";

}

else

{

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

{

cout<<que[i]<<endl;

}

}

}

};

int main()

{

queue q;

int choice;

do

{

cout<<"\n\t\t\t\t\*\*\*\*\*\*\*\*\*\*Queue Using Array\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\n1) For insert Element in Queue :";

cout<<"\n2) For Delete element from Queue :";

cout<<"\n3) For Display Queue Element : ";

cout<<"\n Please Enter Your choice : ";

cin>>choice;

switch(choice)

{

case 1:

q.insert();

break;

case 2:

q.deQueue();

break;

case 3:

q.display();

break;

default:

cout<<"\nInvalid choice..!!!\n";

break;

}

cout<<"\nPress 1 to repeat : ";

cin>>choice;

}while(choice==1);

return 0;}