NAME :- Aindrail Santra

Roll No. :- 2029044

ASSIGNMENT-7

Q 2. Write a menu driven program to implement queue operations such as Enqueue, Dequeue, Peek, Display of elements, IsEmpty, IsFull using static array.

```
#include<iostream.h>
using namespace std;
#define n 100
class Queue{

int* arr;
int front;
int back;
public:
Queue(){
arr = new int[n];
front = -1;
back = -1;
}

void Enqueue(){

int x;
cin>>x;
```

NAME :- Aindrail Santra

Roll No. :- 2029044

```
if(back==n-1){
cout<<"Queue overflow"< return;
}
back++;
arr[back]=x;
if(front==-1){
front++;
}
}
void Dequeue(){
if(front==-1 || front>back){
cout<<"NO elements in queue\n";
return ;
}
front++;
}
void peek(){
if(front==-1 || front>back){
cout<<"NO elements in queue\n";
}
cout< arr[front]<
}
void display(){
int temp = back ;
while(temp>=0){
```

NAME :- Aindrail Santra

Roll No. :- 2029044

```
cout<< temp--;
}
}
bool Isempty(){
if(front==-1 || front>back){
cout<<"Queue is empty\n";
return true;
}
cout<<"Not empty\n";
return false;
}
bool isFull(){
if(front>n){
cout<<"Queue Overflow\n";
return true;
}
cout<<"Not full yet\n";
return false;
}
};
void menu()
{
cout << "Press 1 to enter element / Enqueue \n";
cout << "Press 2 to dequeue \n";
```

NAME :- Aindrail Santra

Roll No. :- 2029044

```
cout << "Press 3 to peek in the queue\n";
cout << "Press 4 to display all elements\n";
cout << "Press 5 to check the stack is empty or
not\n";
cout<<"Press 6 to check the stack is full or not\n";
cout << "Press 7 to exit\n";
}

int main(){
Queue q;
int choice;
do {
cout << "\n";
menu();
cout << "\nEnter your "
<< "choice:\n ";
cin >> choice;
switch (choice) {
case 1:
q.Enqueue();
break;
case 2:
```

NAME :- Aindrail Santra
Roll No. :- 2029044

```
q.Dequeue();
break;
case 3:
q.peek();
break;
case 4:
q.display();
break;
case 5:
q.Isempty();
break;
case 6:
q.isFull();
break;
case 7:
cout<<"Thank You :)\n";
exit(0);
break;
default:
cout << "INVALID CHOICE :-(";
}
} while (choice != 7);
return 0 ;
}
```

NAME :- Aindrail Santra
Roll No. :- 2029044

Q 3. Write a menu driven program to implement queue operations such as Enqueue, Dequeue, Peek, Display of elements, IsEmpty using linked list.

```
#include<iostream>
using namespace std;
class node{
public:
int data ;
node* next;
node( int val){
data = val;
next = NULL;
}
};
class queue{
node* front;
node* back;
public:
queue (){
front = NULL;
back=NULL;
}
void Enqueue(){
```

NAME :- Aindrail Santra
Roll No. :- 2029044

```
int x;
cin>>x;
node* n = new node(x);
if(front==NULL){
back=n;
front = n;
return;
}
back->next = n;
back = n;
}
void Dequeue()
{
if(front==NULL){
cout<<"Queue underflow\n";
return;
}
node* todelete = front;
front= front->next;
delete todelete;
}
int peek(){
if(front==NULL){
cout<<"NO ELMENTS \n";
return -1;
```

NAME :- Aindrail Santra
Roll No. :- 2029044

```
}
return front->data;
}
void display(){
node* temp= front;
while (temp!=NULL)
{
cout<<data<<" ";
temp= temp->next;
}
cout<
}
bool Isempty(){
if(front==NULL){
cout<<"The Queue is empty\n";
return true;
}
cout<<"The Queue is not empty\n";
return false;
}
};
void menu()
{
cout << "Press 1 to enter element / Enqueue \n";
```

NAME :- Aindrail Santra
Roll No. :- 2029044

```
cout << "Press 2 to dequeue \n";
cout << "Press 3 to peek in the queue\n";
cout << "Press 4 to display all elements\n";
cout << "Press 5 to check the stack is empty or
not\n";
cout << "Press 6 to exit\n";
}
int main(){
queue q;
int choice;
do {
cout << "\n";
menu();
cout << "\nEnter your "
<< "choice:\n ";
cin >> choice;
switch (choice) {
case 1:
q.Enqueue();
break;
case 2:
q.Dequeue();
break;
```

NAME :- Aindrail Santra
Roll No. :- 2029044

```
case 3:
q.peek();
break;
case 4:
q.display();
break;
case 5:
q.Isempty();
break;
case 6:
cout<<"Thank You :)\n";
exit(0);
break;
default:
cout << "INVALID CHOICE :-(";
}
} while (choice != 6);
return 0 ;
}
```