DS PRACTICAL NO : 8 A

AIM: Implement a Queue and perform the Queue operations: Enqueue, Dequeue and Print using Menu Driver Program such as 1. Add, 2. Delete and 3. Print and 4. Exit.

PROGRAM ;

```
#include<stdio.h>
int Queue[100];
int front = -1, rear = -1, data;
//FUNCTION FOR ENQUEUE
int enqueue(){
//Checking the queue is full or not
if(rear == 99){
printf("Sorry, The Queue is Overflow!\n");
}else if (front == -1 && rear == -1)
{
printf("Enter the data:\t");
scanf("%d", &data);
//Checking the input element is first or not
front = 0;
rear = 0;
Queue[0] = data;
}else{
printf("Enter the data:\t");
scanf("%d", &data);
rear++;
Queue[rear] = data;
}
return 0;
```

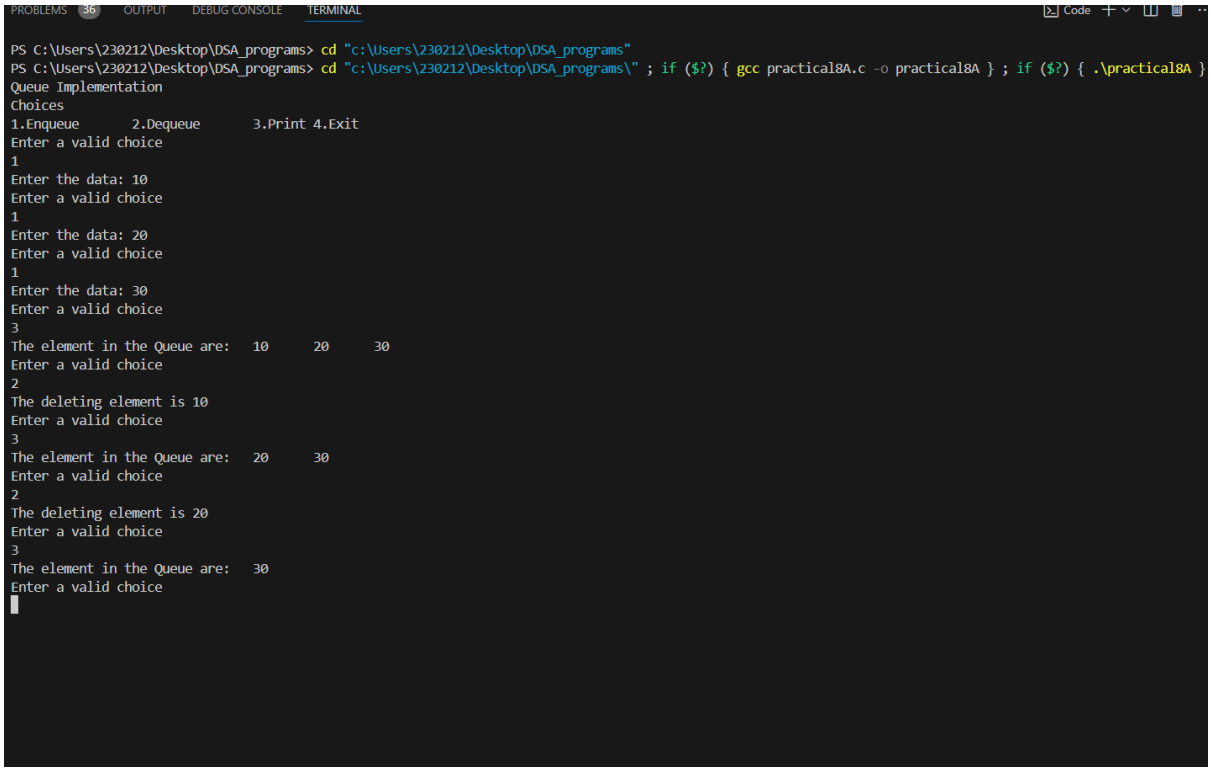
```
}
//FUNCTION FOR DEQUEUE
int dequeue(){
//Checking the Queue is empty or not.
if(front == -1){
printf("The Queue is Empty to delete a element.\n");
}else if(front > rear){
//Checking all the element is deleted or not.
printf("The Queue is Empty to delete a element.\n");
front = -1;
rear = -1;
}else{
//Simply deleting the element from front.
printf("The deleting element is %d\n", Queue[front]);
front++;
}
return 0;
}
void display(){
if(front == -1 || front > rear){
//Checking the queue is empty or not.
printf("The Queue is empty so, can not print the element.\n");
}else{
//printing the elements in the Queue
printf("The element in the Queue are:\t");
for(int i = front; i <= rear; i++){
printf("%d\t", Queue[i]);
}
```

```
printf("\n");


}
}
//MAIN FUNCTION
int main(){
int choice;
printf("Queue Implementation\n");
printf("Choices\n1.Enqueue\t2.Dequeue\t3.Print\t4.Exit\n");
do
{
printf("Enter a valid choice\n");
scanf("%d", &choice);
switch (choice)
{
case 1:
enqueue();
break;
case 2:
dequeue();
break;
case 3:
display();
break;
case 4:
printf("You exited the Program successfully.");
break;
default:
```

DS PRACTICAL NO : 8 A

```
printf("Please enter a valid choice as mention!\n");

break;

}

} while (choice != 4);

return 0;

}
```

OUTPUT



GITHUB LINK : https://github.com/AmolNagargoje04/Data-Structure-practical