PRACTICAL 08 [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>


//Creating array Globaly
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: ");
        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:
printf("Please enter a valid choice as mention!\n");
break;
}
} while (choice != 4);
return 0;
}
```

OUTPUT

```
PS C:\Users\chuna> g++ p8a.c
PS C:\Users\chuna> ./a.exe
Queue Implementation
Choices
1.Enqueue       2.Dequeue       3.Print 4.Exit
Enter a valid choice: 1
Enter the data: 12
Enter a valid choice: 1
Enter the data: 23
Enter a valid choice: 1
Enter the data: 34
Enter a valid choice: 1
Enter the data: 45
Enter a valid choice: 2
The deleting element is 12
Enter a valid choice: 3
The element in the Queue are:   23      34      45
Enter a valid choice: 4
You exited the Program successfully.
PS C:\Users\chuna>
```

GITHUB LINK : https://github.com/Nishikant-Chunarkar/DATA_STRUCTURE_PRACTICAL