CSE 2003: Lab Assignment #3

Due on Thursday, March 1, 2017

Shaik Naseera 2:00pm

Jacob John

Contents

Problem 1

Write a program to implement Linear Queue using Array.

Listing 1: C program to implement Linear Queue using Array

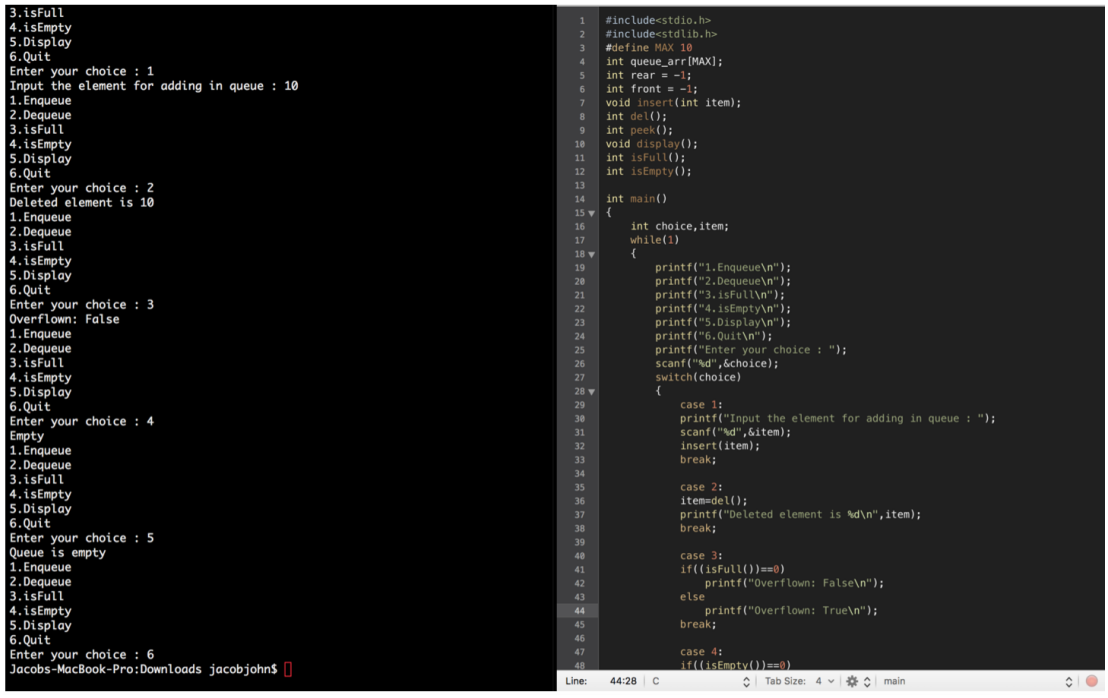
```
#include<stdio.h>
#include<stdlib.h>
#define MAX 10
int queue_arr[MAX];
int rear = -1;
int front = -1;
void insert(int item);
int del();
int peek();
void display();
int isFull();
int isEmpty();

int main()
     {
          int choice,item;
          while(1)
          {
               printf("1.Enqueue\n");
               printf("2.Dequeue\n");
               printf("3.isFull\n");
               printf("4.isEmpty\n");
               printf("5.Display\n");
               printf("6.Quit\n");
               printf("Enter your choice : ");
               scanf("%d",&choice);
               switch(choice)
               {
                    case 1:
                    printf("Input the element for adding in queue : ");
                    scanf("%d",&item);
                    insert(item);
                    break;

                    case 2:
                    item=del();
                    printf("Deleted element is %d\n",item);
                    break;

                    case 3:
                    if((isFull())==0)
                         printf("Overflown: False\n");
                    else
                         printf("Overflown: True\n");
                    break;

                    case 4:
                    if((isEmpty())==0)
```

Problem 1 continued on next page...

```
                        printf("Not Empty\n");
                    else
                        printf("Empty\n");
                    break;

                    case 5:
                    display();
                    break;

                    case 6:
                    exit(1);
                    break;

                    default:
                    printf("Wrong choice\n")
                }/*End of switch*/
            }/*End of while*/
        }/*End of main()*/

void insert(int item)
{
        if(isFull())
            {
                printf("Queue Overflow\n");
                return;
            }
        if(front==-1)
            front=0;
        rear=rear+1;
        queue_arr[rear] = item;
}/*End of insert()*/

int del()
{
        int item;
        if(isEmpty())
            {
                printf("Queue Underflow\n"); exit(1);
            }
        item = queue_arr[front];
        front=front+1;
        return item;
}/*End of del()*/

int isEmpty()
{
        if(front == -1 || front == rear+1)
            return 1;
        else
            return 0;
}/*End of isEmpty()*/

int isFull()
```

```
{
      if(rear == MAX-1)
            return 1;
      else
            return 0;
}/*End of isFull()*/

void display()
{
      int i;
      if(isEmpty())
      {
            printf("Queue is empty\n");
            return;
      }
      printf("Queue is: \n\n");
      for(i=front; i<=rear; i++)
            printf("%d ",queue_arr[i]);
      printf("\n\n");
}/*End of display*/
```

Output:



Problem 2

Write a program to implement Circular Queue using Array.

Listing 2: C program to implement Circular Queue using Array

```c
#include<stdio.h>
#include<stdlib.h>
#define max_size 10
int cqueue[max_size],front=-1,rear=-1;
void insert();
void del();
void display();
int isFull();
int isEmpty();
int main()
{
        int choice;
        do{

                printf("\n\n--------CIRCULAR QUEUE OPERATIONS----------\n");
                printf("1.Enqueue\n");
                printf("2.Dequeue\n");
                    printf("3.isFull\n");
                    printf("4.isEmpty\n");
                printf("5.Display\n");
                printf("6.Exit\n");
                printf("----------------------");
                printf("\nEnter your choice:\t");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                              insert();
                              break;

                              case 2:
                              del();
                              break;

                        case 5:
                              display();
                              break;

                              case 6:
                              exit(0);
                              break;

                              case 3:
                              if((isFull())==0)
                                   printf("Overflown: False\n");
                              else
                                   printf("Overflown: True\n");
                              break;
```

```
                                case 4:
                                if((isEmpty())==0)
                                        printf("Not Empty\n");
                                else
                                        printf("Empty\n");
                                break;

                        default:
                                printf("\nInvalid choice:\n");
                                break;
                }
        }while(choice!=6);

}
void insert() //Inserting an element in to the queue
{
        int item;
        if(front==(rear+1)%max_size)
        {
                printf("\nQueue Overflow:");
        }
        else
        {
                printf("Enter the element to be inserted:\t");
                scanf("%d",&item);
                rear=(rear+1)%max_size;
                cqueue[rear]=item;

                if(front==-1)
                {
                        front=0;
                        rear=0;
                }

        }

}//end of insert()

void del()      //deleting an element from the queue
{
        int item;
        if(front==-1)
        {
                printf("\nQueue Underflow:");
        }
        else
        {
                item=cqueue[front];
                printf("\nThe deleted element: %d\t",item);
                if(front==rear)
                {
                        front=-1;
```

```
                        rear=-1;
                }
                else
                {
                        front=(front+1)%max_size;
                }

        }
}//end of del()

void display()  //To display the queue elements
{
        int i;
        if(front==-1)
        {
                printf("\nQueue is Empty:");
        }
        else
        {
                printf("\nThe queue elements are:\n" );
                if(front<rear)
                {
                        for(i=front;i<=rear;i++)
                        {
                                printf("%d\t",cqueue[i]);
                        }
                }
                else
                {
                        for(i=0;i<=rear;i++)
                        {
                                printf("%d\t",cqueue[i]);
                        }
                        for(i=front;i<max_size;i++)
                        {
                                printf("%d\t",cqueue[i]);
                        }
                }

        }

}//end of display()

int isEmpty()
{
     if (front==-1) return 1;
     else
          return 0;
}/*End of isEmpty()*/

int isFull()
{
     if((front==0 && rear==max_size-1)||(front==rear+1))
```

```
155             return 1;
        else
                return 0;
}/*End of isFull()*/
```

Output: