LAB - 2

024

Date
Page

1Y Write a program to simulate the working of the queue of integers using an array. Provide the following operations : Insert, delete, display. The program should print appropriate message for overflow and underflow condition.

```
#include <stdio.h>
#define MAX 50
int q_array[MAX];
int rear = -1;
int front = -1;

void display()
{
   int i;
   if (front == -1)
      printf ("queue is empty \n")
   else
   {
      printf ("queue is : \n");
      for (i = front; i <= rear; i++)
         printf ("%d", q_array[i]);
      printf("\n");
   }
}
```

```
void insert()
{
   int add_item;
   if (rear == MAX-1)
   printf("queue overflow \n");
   else
   {
      if (front == -1)
         front = 0;
      printf("\nInsert the element in queue:");
      scanf("%d", &add_item);
      rear = rear + 1;
      q_array[rear] = add_item;
   }
}

void delete()
{
   if (front == -1 || front > rear)
   {
      printf("queue underflow \n");
   }
   else
   {
      printf("Deleted element is %d\n", q_array[front]);
      front = front + 1;
   }
}
```

```
int main()
{
    int choice;
    while(1)
    {   printf("\n");
        printf("1.Insert\n");
        printf("2. Delete\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your choice :");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1 : insert();
                     break;
            case 2 : delete();
                     break;
            case 3 : display();
                     break;
            case 4 : exit(1);
            default : printf("Invalid choice\n");
        }
    }
    return 0;
}
```

output :

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert : 10

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice : 1
Enter element to insert : 20

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice : 3
Queue is :
10
20

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice : 2
Deleted element is : 10

2. write a program to simulate the working of a circular queue using an array. provide the following operations : insert, delete & display. The program should print appropriate message for queue empty and queue overflow conditions.

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
int items[SIZE], rear = -1, front = -1;
int isFull()
{
    if ((front == rear+1) || (front == 0 && rear == SIZE-1))
        return 1;
    return 0;
}
int isEmpty()
{
    if (front == -1)
        return 1;
    return 0;
}
```

```
void enqueue(int element)
{
    if (isFull())
    {
        printf("\n queue is full");
    }
    else {
        if (front == -1)
            front = 0;
        rear = (rear+1) % SIZE;
        items[rear] = element;
        printf("%d is inserted", element);
    }
}
int dequeue()
{
    int value;
    if (isEmpty())
    {
        printf("\n queue is empty !! \n");
        return -1;
    }
    else
    {
        value = items[front];
```

```
    if (front == rear)
    {
        front = -1 ;
        rear = -1 ;
    }
    else
        front = (front + 1) % SIZE ;
    return (value) ;
  }
}
void display()
{
    int i ;
    if (isEmpty())
        printf ("\n queue is empty") ;
    else
    {
        printf ("\n front position = %d \n", front) ;
        for (i = front ; i != rear ; i = (i+1)% SIZE)
        {
            printf ("\t %d \t", items[i]) ;
        }
        printf ("%d \t", items[i]) ;
    }
}
```

```
void main ()
{
    int choice, element ;
    while(1)
    {
        printf("\n 1. insert \n 2. delete \n 3. display \n 4. exit");
        printf("\n enter choice:") ;
        scanf("%d", &choice");
        switch (choice)
        {
            case 1 : printf("\n enter the element to insert:");
                     scanf("%d", &element);
                     enqueue (element) ;
                     break ;
            case 2 : element = dequeue () ;
                     if (element != -1)
                         printf("%d element is deleted",
                                element) ;
                     break ;
            case 3 : display () ;
                     break ;
            case 4 : exist(0) ;
            default : printf ("\n invalid choice");
        }
}
```

Output :

1. insert
2. delete
3. display
4. exit

Enter choice : 1

Enter element to insert : 5

5 is inserted

1. insert
2. delete
3. display
4. exit

Enter choice : 1

Enter element to insert : 10

10 is inserted

08/01/24