Lab 4: Circular queue.

WAP to simulate the working of circular queue of integers using an array. Provide the following operation:
a) Insert b) Delete c) Display.

```
#include<stdio.h>
#include<stdlib.h>
#define SIZE 3.
int item;
int front=0, rear=-1, q[size], count=0;
void insertcir()
{  if(count==SIZE)
   {  printf("Queue overflow\n");
   } return;
   rear=(rear+1)%SIZE;
   q[rear]=item;
   printf("%d inserted successfully\n", item);
   count++;
}
int deletefront()
{  if(count==0)
   return -1;
   item=q[front];
   front=(front+1)%SIZE;
   count=count-1;
   return item;
}
```

```
void display()
{
  int i,f;
  if(count==0)
  { printf("Queue is empty\n");
  } return;
  f=front;
  printf("Contents of queue :");
  for(i=1;i<=count;i++)
  { printf("%d ",q[f]);
    f=(f+1)%SIZE;
  }
  printf("\n");
}
void main()
{ int choice;
  for(;;)
  { printf("\n1. Insert 2. delete 3. display\n");
    printf("Enter choice :");
    scanf("%d",&choice);
    switch(choice)
    { case 1: printf("Enter item to be inserted:");
              scanf("%d",&item);
              insertrear();
              break;
```

```
case 2 : item = deletefront();
         if (item == -1)
             printf("Queue is empty\n");
         else
             printf("Item deleted : %d\n", item);
         break;
case 3 : display();
         break;
default : exit(0);
    }
  }
}
```

Output:

1. Insert 2. delete 3. display
Enter choice : 1
Enter the item to be inserted : 10

1. Insert 2. delete 3. display
Enter choice : 1
Enter the item to be inserted : 20

1. Insert 2. delete 3. display
Enter choice : 1
Enter the item to be inserted : 30

Circular Queue.

1. Insert 2. delete 3. display
Enter choice : 1
Enter the item to be inserted : 40
Queue overflow

1. Insert 2. delete 3. display
Enter choice : 2
Item deleted : 10.

1. Insert 2. delete 3. display
Enter choice : 1
Enter the item to be inserted : 5.

1. Insert 2. delete 3. display
Enter choice : 3.
Contents of queue : 20 30 5

1. Insert 2. delete 3. display
Enter choice : 4