(3) WAP to simulate the working a circular queue to integers using an array provide the following operations

a) Insert b) Delete c) Display

```
#include <stdio.h>
int rear=-1;
int front=-1;
int max=5;
void Enqueue(int arr[], int value){
  if (rear==-1 || front==-1){
      rear++;
      front++;
      arr[rear]=value;
      rear++;}
else if (rear==max){
    if (front!=0){
        rear=0;
      arr[rear]=value;
      rear++;}
  else{
    Print("overflow\n");
  }
}
else if (rear==(front)){
      Printf("overflow");}
else{
    arr[rear]=value;
    rear++;}}
void Dequeue(int arr[], int value){
  if (front==-1 || rear==-1){
      Printf("underflow\n");}
else if (front==(rear-1)){
  Printf("Deleted element=%d\n",arr[front]);
  rear=-1;
  front=-1;}
```

```
    else {
    int temp = arr[front];
    front++;
    Printf("Deleted element = %d\n", temp);
}}

int main(){
    int choice;
    int arr[5];
    int value;

void operations(){
Printf("Enter appropriate number to perform operations:
\n1. Enqueue \n2. Dequeue \n3 Display \n4. Exit\n");
    scanf("%d", &choice);
    switch(choice){
        case 1:
            Printf("Enter the value to insert \n");
            scanf("%d", &value);
            Enqueue(arr, value);
            operations();
            break;

    case 2:
            Dequeue(arr);
            operations();
            break;

    Case 3:
            display(arr);
            operations();
            break;
```

```
case 4:
    Printf(" Exited \n");
    break;
default:
    Printf("Invalid choice \n");
    operations();
    break; }}
operations();
return 0;
}
```

output:

Enter appropriate number to perform operations:

1. Enqueue
2. Dequeue
3. Display
4. Exit

1.

Enter the value to insert

10

Enter the value to insert

20

Enter the value to insert

30

Enter the value to insert

40

Enter the value to insert

50

Enter the value to insert

60

overflow

Deleted element = 10

Enter the value to insert

100

100 20 30 40 50

2/1/24