Date: 8/1/2024

1. Write a program to simulate the working of the of integers ~~xxx~~ using an array. Provide the following operation: Insert, delete, display. The program should print appropriate message for overflow & underflow conditions.

```
#include <stdio.h>
void insert (int front, int rear);
{
  int num;
  printf("\nenter number: ");
  scanf("%d", num);
void delete();
void display();
int
#include<stdlib.h>
#include<stdio.h>
#define MAX 50
void insert();
int delete();
void display();
int arr[MAX];
int rear = -1;
int front = -1;
void main()
{
  int ch,choice;
  while(1)
  {
    printf("1. Insert element to queue \n");
    printf("2. Delete element from queue \n");
    printf("3. Display all element of queue\n");
    printf("4. Quit \n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
```

```
switch(choice)
{
    case 1: insert();
            break;
    case 2: delete(); p= delete(); printf("Deleted:%d", p);
            break;
    case 3: display()
            break;
    case 4: exit(1);
    default: printf("wrong choice.");
}
}
void insert() {
int item;
if(rear == MAX-1)
printf("Overflow \n");
else {
if(front == -1)
front = 0;
printf("Enter element: ");
scanf("%d", &item);
rear = rear+1;
arr[rear] = item;
}}
int delete()
if(front == -1 || front > rear) {
printf("Underflow.");
return ; }
else {
front = front+1;
return arr[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", que[i]);
printf("\n"); }}
```

```
Output
1 enter element to queue
2 delete element from que
3 display all elements from queue
4 quit
enter your choice 1
enter element: 3
1 enter element to queue
2 delete element from queue
3 display all elements from queue 3
```

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

```
#include<stdlib.h>
#include<stdio.h>
#define max 5
int front=-1, rear=-1
int que[max];
void insert();
int delete();
void display();
int main(){
int choice, no;
while(1){
printf("\n1. Insert");
printf("\n2. Delete");
printf("\n3. Display");
printf("\n4. Exit");
printf("Enter choice: ");
scanf("%d", &choice);
```

```
switch(choice) {
case 1: insert();
        break;
case 2: no= delete();
        break;
case 3: display();
        break;
case 4: exit(1);
default: printf("\n Wrong choice \n");
}}}

void insert() {
  int no;
  if ((front==0 && rear == max-1)|| front== rear+1) {
      printf("\n Overload ");
      return;
  }
  printf("\nEnter a number to insert: ");
  scanf("%d", &no);
  if (front == -1)
      front = front+1;
  if (rear == max-1)
     rear=0;
  else rear = rear+1;
  arr[rear]= no;
}

int delete() {
int e;
if(front == -1) {
     printf("Underflow."); }
e= arr[front]
if (front == max-1) front= 0;
```

```
else if (front == rear) {
    front = -1;
    rear = -1; }
else front = front + 1
printf("\n%d was dlted! \n", e);
return e;
}
void display() {
int i;
if (front == -1) {
    printf("\n Empty list; ");
    return; }
i = front;
if (front <= rear) {
printf("\n\n");
while(i <= rear) printf("%d", arr[i++]);
printf("\n"); }
else {
printf("\n\n");
while (i <= max-1) printf("%d", arr[i++]);
i = 0;
while(i <= rear) printf("%d", arr[i++]);
printf("\n"); } }
```

O/P:

1. insert
2. dlte
3. Display
4. Exit

Enter Choice : 1

3

1. ~~insert~~
2. dlete
3. Display

4. Exit

3

3

8/01/24