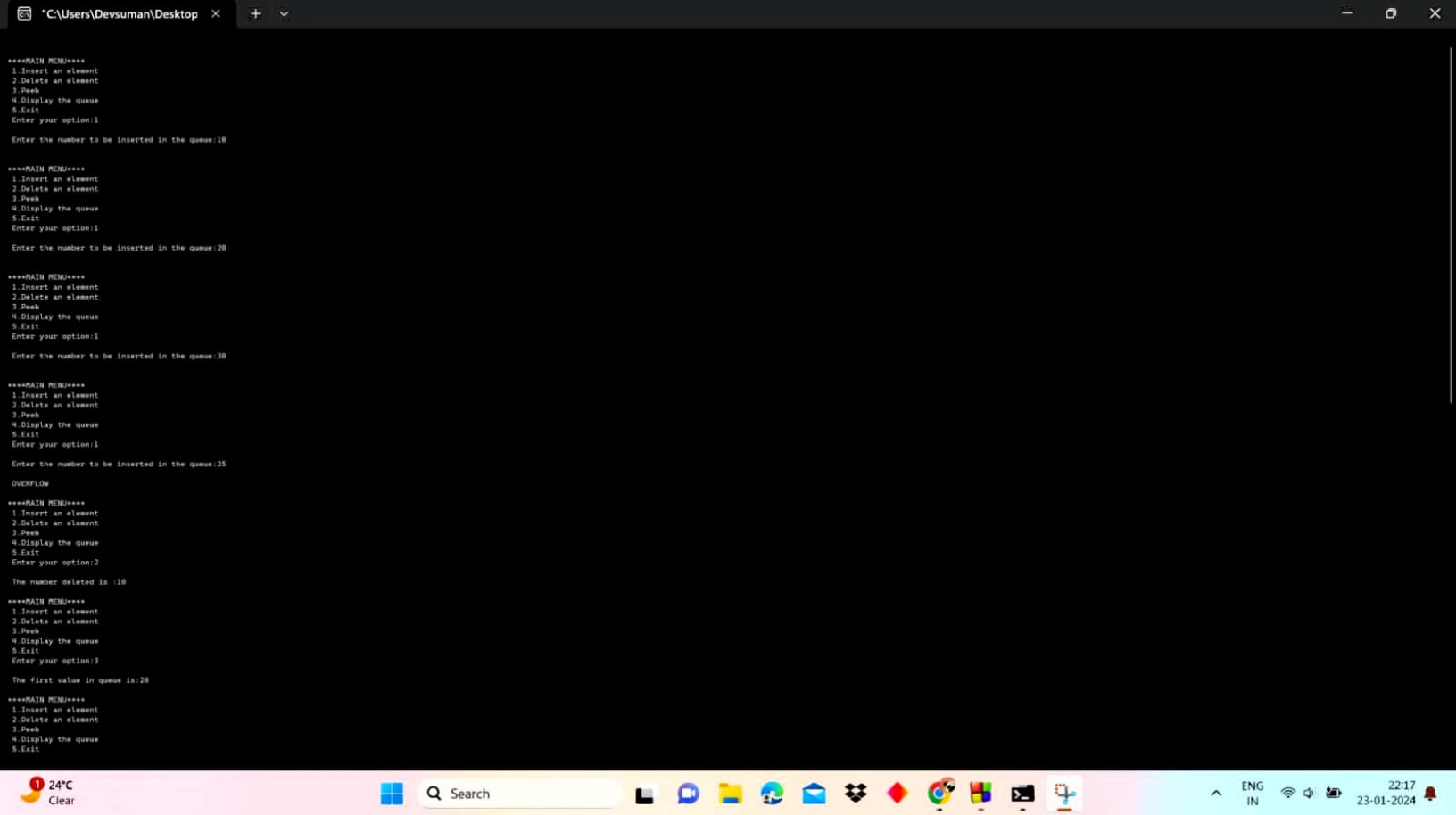
08/01/24

Q3. 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 ~~be~~ print appropriate message for overflow and underflow condition.

```
#include <stdio.h>
#include <conio.h>
# define MAX 10
int queue [MAX];
int front = -1, rear = -1;
void insert (void);
int delete_element (void);
int peek ( void);
void display (void);
int main()
{
  int option, val;
  do
  {
    printf("\n\n ***** MAIN MENU *****");
    printf("\n 1. Insert an element");
    printf("\n 2. Delete an element");
    printf("\n 3. Peek");
    printf("\n 4. Display the queue");
    printf("\n 5. Exit");
    printf("\n Enter your option:");
    scanf("%d", &option);
    switch(option)
    {
      case 1:
        insert();
        break;
      case 2:
        val = delete_element();
        if(val != -1)
```

```
        printf("\n The number is deleted is : %d", val);
        break;
      case 3:
        val = peek();
        if (val != -1)
          printf("\n The first value in queue is : %d", val);
        break;
      case 4:
        display();
        break;
      }
    }
    while (option != 5);
    getch();
    return 0;
}
void insert()
{
    int num;
    printf("\n Enter the number to be inserted in the
    queue : ");
    scanf("%d", &num);
    if (rear == MAX-1)
      printf("\n OVERFLOW");
    else if (front == -1 && rear == -1)
      front = rear = 0;
    else
      rear++;
    queue[rear] = num;
}
int delete_element()
{
    int val;
    if (front == -1 || front > rear)
```

```
    {
        printf("\n UNDERFLOW");
        return -1;
    }
    else
    {
        val = queue[front];
        front++;
        if (front > rear)
            front = rear = -1;
        return val;
    }
}
int peek()
{
    if (front == -1 || front > rear)
    {
        printf("\n QUEUE is EMPTY");
        return -1;
    }
    else
    {
        return queue[front];
    }
}
void display()
{
    int i;
    printf("\n");
    if (front == -1 || front > rear)
        printf("\n QUEUE is EMPTY");
    else
    {
        for(i = front; i <= rear; i++)
            printf("\t %d", queue[i]);
    }
}
```


"C:\Users\Devsuman\Desktop
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
Enter the number to be inserted in the queue:10
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
Enter the number to be inserted in the queue:20
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
Enter the number to be inserted in the queue:30
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
Enter the number to be inserted in the queue:25
OVERFLOW
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:2
The number deleted is :10
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:3
The first value in queue is:20
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
24°C
Clear
Search
ENG
IN
22:17
23-01-2024

Q4. 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 <conio.h>
#include <process.h>
#define QUE_SIZE 3
int item, front=0, rear=-1, q[QUE_SIZE], count=0;
void insertrear()
{
  if (count== QUE_SIZE)
  {
     printf ("Queue overflow\n");
     return;
  }
  rear = (rear+1) % QUE_SIZE;
  q[rear] = item;
  count++;
}
int deletefront();
{
  if (count==0)
  return -1;
  item = q[front];
  front = (front+1)% QUE_SIZE;
  count = count-1;
  return item;
}
void display()
{
  int i,f;
  If(count ==0)
```

```
{
    printf(" Queue is empty \n");
    return;
}
i = front;
printf(" contents of queue \n");
for(i=1; i<=count; i++)
{
    printf("%d \n", q[i]);
    i = (i+1) % QUE_SIZE;
}
}
void main()
{
  int choice;
  clrscr();
  for(;;)
  {
    printf("\n 1. insert rear \n 2. delete front \n 3. display
    \n 4. exit \n");
    printf(" Enter the choice \n");
    scanf("%d", &choice);
    switch( choice)
    {
      case 1: printf(" Enter the item to be inserted \n");
              scanf("%d", &item);
              insert rear();
              break;

      case 2: item = delete front();
              if( item == -1)
              printf("Queue is empty. \n");
              else
              printf(" Item deleted = %d \n", item);
```

```
    break;
  case 3: display();
          break;
  default: exit(0);
 }
}
```

O/P:- Enter
1. Insertrear
2. delete front
3. display
4. Exit 2
Underflow

Enter
1. Insertrear
2. deletefront
3. display
4. Exit 1
Enter value 9
value inserted

Enter
1. Insertrear
2. deletefront
3. display
4. Exit 1
Enter value:
3
value inserted

Enter
1. Insertrear
2. deletefront
3. display
4. Exit 1
Enter value 12
OVERFLOW

Enter
1. insertrear
2. delete front
3. display
4. Exit 1
Enter value 6
value inserted

Enter
1. Insertrear
2. deletefront
3. display
4. Exit 1
3
6
9

Enter
1. Insertrear
2. deletefront
3. display
4. Exit 2
6
9

```
"C:\Users\Devsuman\Desktop

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
10

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
20

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =10

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
3
Contents of queue
20

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
4

Process returned 0 (0x0)   execution time : 22.206 s
Press any key to continue.
```