QUEUE:

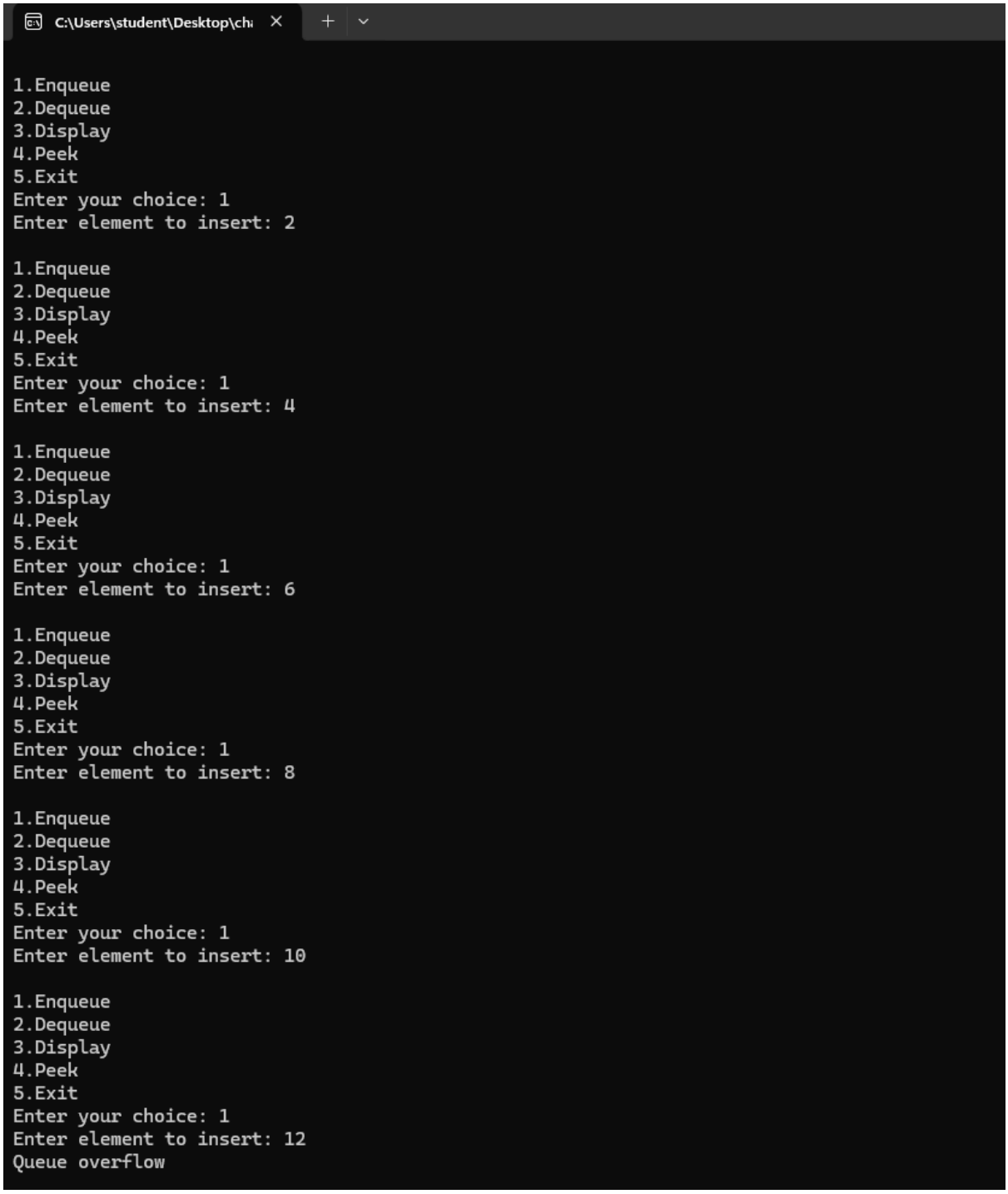
```
#include <stdio.h>
#define N 5

int queue[N];
int front=-1,rear=-1;

void enqueue(int x) {
if (rear==N-1) {
printf("Queue overflow\n");
}
else if(front==-1 && rear==-1) {
front=rear=0;
queue[rear]=x;
}
else {
rear++;
queue[rear]=x;
}
}

void dequeue() {
if (front==-1 && rear==-1){
printf("Queue is empty\n");
}
else if(front==rear){
printf("Deleted element is:  %d\n",queue[front]);
front=rear=-1;
}
```

```
else{
printf("Deleted element is: %d\n",queue[front]);
front++;
}
}

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

void peek(){
if (front==-1 && rear==-1){
printf("Queue is empty\n");
}
else{
printf("Front element: %d\n",queue[front]);
}
}

int main() {
int choice,x;
```

```
do{
printf("\n1.Enqueue\n");
printf("2.Dequeue\n");
printf("3.Display\n");
printf("4.Peek\n");
printf("5.Exit\n");
printf("Enter your choice: ");
scanf("%d",&choice);

switch(choice) {
case 1:
   printf("Enter element to insert: ");
   scanf("%d",&x);
   enqueue(x);
   break;

case 2:
   dequeue();
   break;

case 3:
   display();
   break;

case 4:
   peek();
   break;

case 5:
   printf("Exiting....\n");
   break;
```

```
default:
    printf("Invalid Choice\n");
}
}
while (choice !=5);
return 0;
}
```

OUTPUT:


C:\Users\student\Desktop\ch
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 1
Enter element to insert: 2
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 1
Enter element to insert: 4
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 1
Enter element to insert: 6
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 1
Enter element to insert: 8
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 1
Enter element to insert: 10
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 1
Enter element to insert: 12
Queue overflow

```
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 3
Queue elements are:
2 4 6 8 10

1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 4
Front element: 2

1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 2
Deleted element is: 2

1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 2
Deleted element is: 4

1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 4
Front element: 6
```

```
1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 2
Deleted element is: 8

1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 2
Deleted element is:  10

1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 2
Queue is empty

1.Enqueue
2.Dequeue
3.Display
4.Peek
5.Exit
Enter your choice: 5
Exiting....

Process returned 0 (0x0)   execution time : 63.818 s
Press any key to continue.
```

OBSERVATION:

13/10/25

Program-3

WAP to simulate the working of a queue of integers using an array. Provide the following operations: Insert, Delete, Display, the program should print appropriate messages for queue empty & queue overflow conditions. Write the pseudocode.

=> Pseudocode:

1. Display an array of size N.
2. Declare Queue [N]
3. Set front to -1.
4. Set rear to -1.
5. If rear = N-1, print "Queue Overflow".
6. Else if FRONT = -1 & REAR = -1, set front = 0, rear = 0 & insert x at Queue [rear].
7. Else, increase rear by 1 & insert x at Queue [rear].
8. End if.
9. If front = -1 & rear = -1, print "Queue is empty".
10. Else if front = rear.
11. print "Deleted element: Queue [front]" and set front = rear = -1.
12. Else, print "Deleted element: Queue [front]" & increase front by 1.
13. End if.

Date __/__/__
Page _____

Display.

13. If front = -1 & rear = -1, print "Queue is Empty"
14. Else, print "Queue elements are:" for i = front to rear, print Queue[i].
15. End if

Peek

16. If front = -1 & rear = -1, print "Queue is Empty"
17. Else, print "Front element: Queue[front]"
18. End if.

19. Repeat menu choices until user selects Exit.
20. Stop the program.

MG

Program:-

```
#include <stdio.h>
#define N 5

int queue[N];
int front = -1, rear = -1;

void enqueue (int x)
{
  if (rear == N-1)
  {
    printf("Queue overflow\n");
  }
else if (front == -1 && rear == -1)
  {
```

```
front = rear = 0;
queue[rear] = x;
}
else {
rear++;
queue[rear] = x;
}
}

void dequeue()
{
  if (front == -1 && rear == -1)
  {
    printf("Queue is empty\n");
  }
  else if (front == rear)
  {
    printf("Deleted element is: %d\n",
                          queue[front]);
    front = rear = -1;
  }
  else
  {
    printf("Deleted element is: %d\n",
                queue[front]);
    front++;
  }
}

void display()
{
  if (front == -1 && rear == -1)
  {
```

```
    printf("Queue is empty\n");
}
else {
    printf("Queue elements are:\n");
    for(int i=front; i<=rear; i++)
    {
        printf("%d ", queue[i]);
    }
    printf("\n");
}
}

void peek()
{
    if(front==-1 && rear==-1)
    {
        printf("Queue is empty\n");
    }
    else {
        printf("Element : %d\n", queue[front]);
    }
}

int main()
{
    int choice, x;

do {
    printf("\n1. Enqueue\n");
    printf("2. Dequeue\n");
    printf("3. Display\n");
    printf("4. Peek\n");
```

Date __/__/__
Page _____

```
        printf ("5. Exit\n");
        printf ("Enter your choices: ");
        scanf ("%d", &choice);

    switch (choice)
    {
        case 1:
            printf ("Enter element to insert: ");
            scanf ("%d", &x);
            enqueue (x);
            break;

        case 2:
            dequeue ();
            break;

        case 3:
            display ();
            break;

        case 4:
            peek ();
            break;

        case 5:
            Exit printf ("Exiting... \n");
                 break;

        default:
            printf ("Invalid Choice\n");
    }
}
while (choice != 5);
return 0;
}
```

Date __/__/__
Page _____

Output :

```
1. Enqueue
2. Dequeue
3. Display
4. Peek
5. Exit.
Enter your choice : 1
Enter element to insert : 2

1. Enqueue
2. Dequeue
3. Display
4. Peek
5. Exit
Enter your choice : 1
Enter element to insert : 4

1. Enqueue
2. Dequeue
3. Display
4. Peek
5. Exit
Enter your choice : 1
Enter element to insert : 6

1. Enqueque
2. Dequeque
3. Display
4. Peek
5. Exit
Enter your choice : 1
Enter element to insert : 8
```

Date __/__/__
Page _____

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

Enter your choice : 1
Enter element to insert : 10

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

Enter your choice : 1
Enter element to insert : 12
Queue Overflow

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

Enter your choice : 3
Queue elements are :
2 4 6 8 10

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

Enter your choice : 4
Front element is : 2

Date __/__/__
Page _____

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

Enter your choice : 2
Deleted element is : 2

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

Enter your choice : 2
Deleted element is : 4

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

Enter your choice : 2
Deleted element is : 6

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

Enter your choice : 2
Deleted element is : 8

Date __/__/__
Page

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

Enter your choice : 2
Deleted element is : 10

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

Enter your choice : 2
Queue is Empty

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

Enter your choice : 5
Exiting....