Exp-no.:5 Name: M.Rohith
Date: 28-11-2022 3122 21 5001 085

SSN College of Engineering

Department of Computer Science and Engineering

UCS2312 – Data Structures Lab

II Year CSE - B Section (III Semester)

Academic Year 2022-23

Staff Incharge: Dr.H. Shahul Hamead

Exercise-5: Exercises on Queues

Aim:

To implement C program in Data structures using the concept of queues and arrays.

Basic

Implement Array version of queues. Make sure queue elements get shifted when dequeuing occurs.

Exp-no.:5
Date: 28-11-2022

Name: M.Rohith
3122 21 5001 085

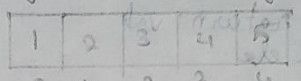
Pseudocode:

28.11.22.

5. Exercises on Queues.

Basic:-

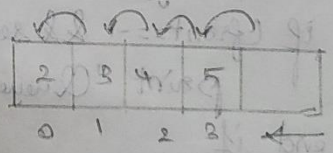
Implement Array version of queues. Make sure queue elements get shifted when dequeuing occurs.



Algorithm:-

Enqueue()
Input: int queue[], int val, front=-1, rear=-1, N=5
Return type: void;

```
if (rear == N-1)
        print "Overflow"
end if
else if (front==-1 && rear==-1)
        front = rear = 0
        queue[rear] = val
end else if
else
        rear = rear + 1
        queue[rear] = val
end else
```



Dequeue()
Return type: int val
Variable: val.

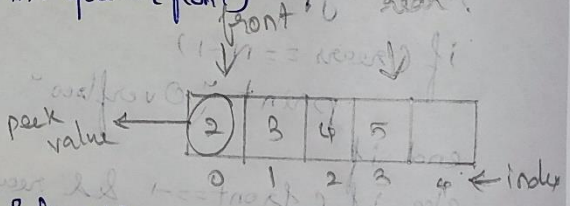
```
if (front==-1 && rear==-1)
    return -1
end if
else if (rear==0)
    val = queue[0]
    rear = front = -1
    return val
end else if
```

Name: M.Rohith
3122 21 5001 085

```
else
    val = queue[0]
    for i=0 to rear-1
        queue[i] = queue[i+1]
    end for
    rear = rear-1
    return val
end else.
```

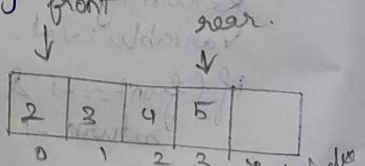
Peek()
Input: int queue[], int front, rear.
Return type: ~~void~~ int queue[front]

```
if (front==-1)
    return -1
end if
else
    return queue[front]
end else.
```



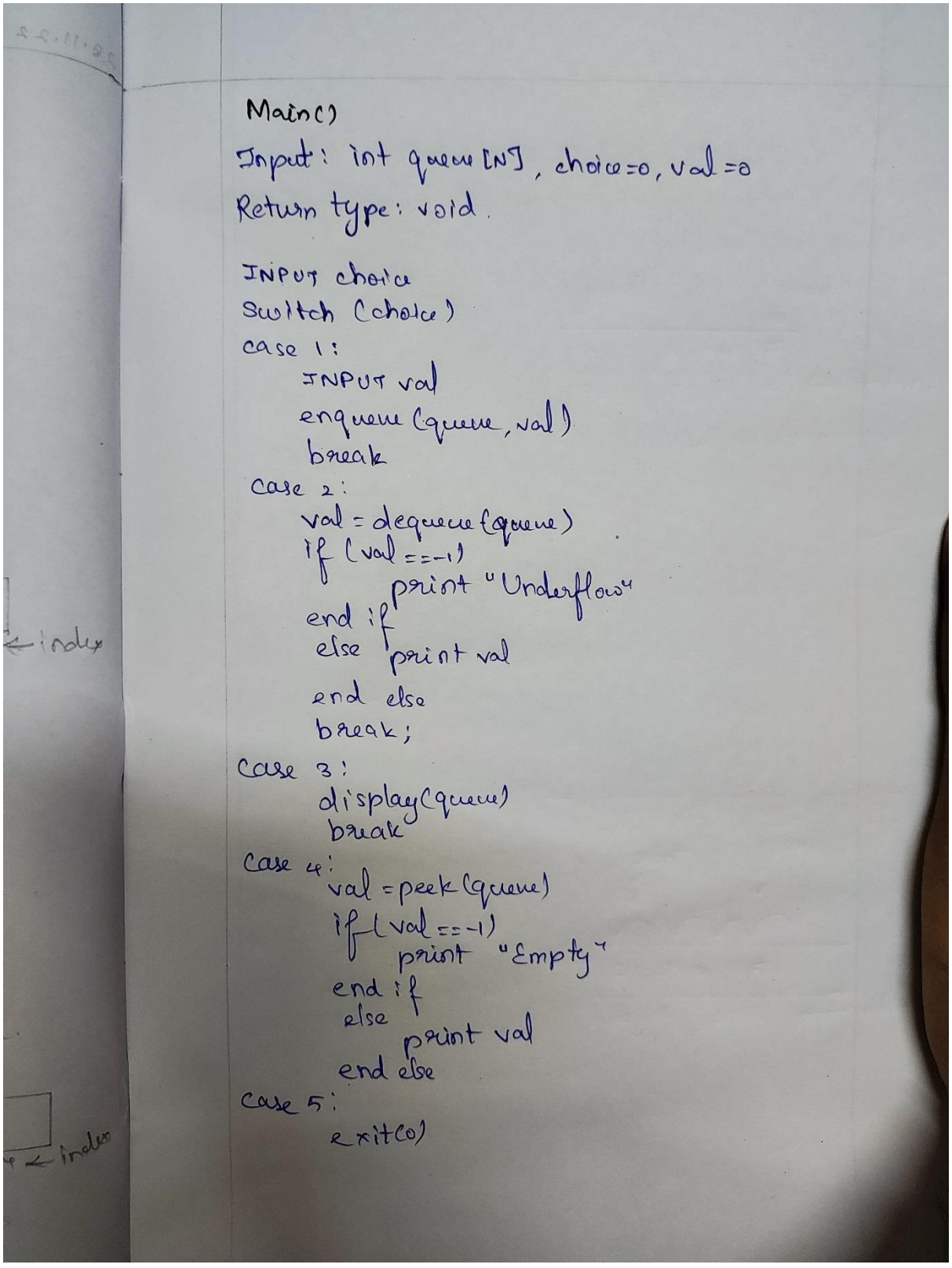
Display().
Input: int queue[], front, rear
Return type: void.

```
if (front==-1 && rear==-1)
    print "Queue is Empty"
end if
else
    for i = front to rear
        print queue[i]
    end for
end else
```



Exp-no.:5
Date: 28-11-2022

Name: M.Rohith
3122 21 5001 085



```
Main()
Input: int queue[N], choice=0, val=0
Return type: void.

INPUT choice
Switch (choice)
case 1:
    INPUT val
    enqueue (queue, val)
    break
case 2:
    val = dequeue (queue)
    if (val == -1)
        print "Underflow"
    end if
    else print val
    end else
    break;
case 3:
    display(queue)
    break
case 4:
    val = peek (queue)
    if (val == -1)
        print "Empty"
    end if
    else
        print val
    end else
case 5:
    exit()
```

Exp-no.:5
Date: 28-11-2022

Name: M.Rohith
3122 21 5001 085

Program code:

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

int front=-1;
int rear=-1;

void enqueue(int queue[],int val)
{
    if(rear==N-1)
        printf("\nOverflow\n");
    else if(front==-1 && rear==-1)
    {
        front=rear=0;
        queue[rear]=val;
        printf("\n%d added to Queue\n",val);
    }
    else
    {
        rear++;
        queue[rear]=val;
        printf("\n%d added to Queue\n",val);
    }
}

int dequeue(int queue[])
{
    int val;
    if(front==-1 && rear==-1)
        return -1;
    else if(rear==0)
    {
        val=queue[0];
        rear=front=-1;
        return val;
    }
    else
    {
        val=queue[0];
```

Name: M.Rohith
3122 21 5001 085

```
        for(int i=0;i<rear;i++)
            queue[i]=queue[i+1];
        rear=rear-1;
        return val;
    }
}

int peek(int queue[])
{
    if(front==-1)
        return -1;
    else
        return queue[front];
}

void display(int queue[])
{
    printf("\nDisplaying values\n");
    if(front==-1 && rear==-1)
        printf("\nQueue is Empty\n");
    else
    {
        for(int i=front;i<=rear;i++)
            printf("%d ",queue[i]);
        printf("\n");
    }
}

void main()
{
    int queue[N],choice=0,val=0;
    do
    {
        printf("\n---------------Menu----------------\n");
        printf("\n1.Enqueue\n2.Dequeue\n3.Display\n4.Peek\n5.Exit\n"
);
        printf("\nEnter the choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                printf("\nEnter the data you want to insert: ");
```

Name: M.Rohith
3122 21 5001 085

```
                scanf("%d",&val);
                enqueue(queue,val);
                break;
            case 2:
                val=dequeue(queue);
                if(val==-1)
                    printf("\nUnderflow\n");
                else
                    printf("\nDequeued value is: %d\n",val);
                break;
            case 3:
                display(queue);
                break;
            case 4:
                val=peek(queue);
                if(val==-1)
                    printf("\nQueue is Empty\n");
                else
                    printf("\nPeek value is: %d\n",val);
                break;
            case 5:
                exit(0);
            default:
                printf("\nInvalid Choice\n");
        }
    } while (choice!=5);
}

/*

front              rear
  0   1   2   3   4
  10  20  30  40  50

front - deletion
rear -  insertion

*/
```

Name: M.Rohith
3122 21 5001 085

Output:

```
PS C:\Rohith\Backup\Desktop\SEM 3\Data Structures in C\Assignment-5> gcc 1.c -o run
PS C:\Rohith\Backup\Desktop\SEM 3\Data Structures in C\Assignment-5> ./run

---------------Menu-----------------

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

Enter the choice: 1

Enter the data you want to insert: 10

10 added to Queue

---------------Menu-----------------

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

Enter the choice: 1

Enter the data you want to insert: 20

20 added to Queue
```

```
---------------Menu-----------------

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

Enter the choice: 1

Enter the data you want to insert: 30

30 added to Queue

---------------Menu-----------------

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

Enter the choice: 1

Enter the data you want to insert: 40

40 added to Queue

---------------Menu-----------------

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

Enter the choice: 1
```

Exp-no.:5 Name: M.Rohith
Date: 28-11-2022 3122 21 5001 085

```
Enter the data you want to insert: 50

50 added to Queue

---------------Menu-----------------

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

Enter the choice: 1

Enter the data you want to insert: 60

Overflow

---------------Menu-----------------

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

Enter the choice: 3

Displaying values
10 20 30 40 50
```

```
---------------Menu-----------------

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

Enter the choice: 4

Peek value is: 10

---------------Menu-----------------

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

Enter the choice: 2

Dequeued value is: 10

---------------Menu-----------------

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

Enter the choice: 4

Peek value is: 20
```

Exp-no.:5 Name: M.Rohith
Date: 28-11-2022 3122 21 5001 085

```
---------------Menu-----------------

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

Enter the choice: 3

Displaying values
20 30 40 50

---------------Menu-----------------

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

Enter the choice: 2

Dequeued value is: 20

---------------Menu-----------------

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

Enter the choice: 2

Dequeued value is: 30
```

```
---------------Menu-----------------

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

Enter the choice: 3

Displaying values

Queue is Empty

---------------Menu-----------------

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

Enter the choice: 2

Underflow

---------------Menu-----------------

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

Enter the choice: 1

Enter the data you want to insert: 60
```

Exp-no.:5 Name: M.Rohith
Date: 28-11-2022 3122 21 5001 085

```
---------------Menu-----------------

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

Enter the choice: 2

Dequeued value is: 40

---------------Menu-----------------

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

Enter the choice: 2

Dequeued value is: 50

---------------Menu-----------------

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

Enter the choice: 4

Queue is Empty
```

```
60 added to Queue

---------------Menu-----------------

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

Enter the choice: 3

Displaying values
60

---------------Menu-----------------

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

Enter the choice: 4

Peek value is: 60

---------------Menu-----------------

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

Enter the choice: 0
```

```
Invalid Choice

---------------Menu-----------------

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

Enter the choice: 5
PS C:\Rohith\Backup\Desktop\SEM 3\Data Structures in C\Assignment-5>
```

Results:

Hence Data Structures in C language has been implemented using the concepts of queues and arrays successfully and the output has been obtained.