**TITLE 39**

Write a C program to implement a Linear-Queue, user must choose the following options:

a. Add an element to the Queue – EnQueue.

b. Remove an element from the Queue – DeQueue.

c. Display the elements of the Queue.

d. Terminate the program.

**OBJECTIVE:**

By the end of this problem we will be able to implement Linear-Queue

**PROBLEM STATEMENT:**

In this program we will be implementing Linear-Queue with the user choosing one of the above options in the question. Input from user:

Enter your choice:

Once the input is collected and stored, the output is displayed.

**ALGORITHM:**

START

Define variables: choice

INPUT: Read from the user

COMPUTATION: Computing the implementation of Linear-Queue

DISPLAY: Displaying the output according to the user’s choice

STOP

**PROGRAM:**

#include <stdio.h>

#define MAX 50

void insert();

void delete();

void display();

int queue\_array[MAX];

int rear = - 1;

int front = - 1;

main()

{

int choice;

while (1)

{

printf("1.Insert element to queue \n");

printf("2.Delete element from queue \n");

printf("3.Display all elements of queue \n");

printf("4.Quit \n");

printf("Enter your choice : ");

scanf("%d", &choice);

switch (choice)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(1);

default:

printf("Wrong choice \n");

} /\* End of switch \*/

} /\* End of while \*/

} /\* End of main() \*/

void insert()

{

int add\_item;

if (rear == MAX - 1)

printf("Queue Overflow \n");

else

{

if (front == - 1)

/\*If queue is initially empty \*/

front = 0;

printf("Inset the element in queue : ");

scanf("%d", &add\_item);

rear = rear + 1;

queue\_array[rear] = add\_item;

}

} /\* End of insert() \*/

void delete()

{

if (front == - 1 || front > rear)

{

printf("Queue Underflow \n");

return ;

}

else

{

printf("Element deleted from queue is : %d\n", queue\_array[front]);

front = front + 1;

}

} /\* End of delete() \*/

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 ", queue\_array[i]);

printf("\n");

}}

**CONCLUSION:**

The simulation of the above program helped me learn about how Linear-Queue is implemented according to the user’s choices.

**OUTPUT:**

1.Insert element to queue

2.Delete element to queue

3.Display all elements of queue

4.Quit

Enter your choice : 1

Insert the element in queue : 4

1.Insert element to queue

2.Delete element to queue

3.Display all elements of queue

4.Quit

Enter your choice : 1

Insert the element in queue : 6

1.Insert element to queue

2.Delete element to queue

3.Display all elements of queue

4.Quit

Enter your choice : 1

Insert the element in queue : 8

1.Insert element to queue

2.Delete element to queue

3.Display all elements of queue

4.Quit

Enter your choice : 3

Queue is :

4 6 8

1.Insert element to queue

2.Delete element to queue

3.Display all elements of queue

4.Quit

Enter your choice : 4