

OUTPUT

Linked List Queue

Choose an operation:

[1] Enqueue

[2] Dequeue

[3] Show Queue

[4] Exit

Choice: 3

Queue Empty!!

Choice: 1

Element to Enqueue: 10

Choice: 1

Element to Enqueue: 20

Choice: 1

Element to Enqueue: 30

Choice: 1

Element to Enqueue: 40

Choice: 3

FRONT

10	20	30	40
----	----	----	----

 REAR

Choice: 2

GOT: 10

Choice: 3

FRONT

20	30	40
----	----	----

 REAR

Choice: 2

GOT: 20

Choice: 3

FRONT

30	40
----	----

 REAR

Choice: 2

GOT: 30

Choice: 3

FRONT

40

 REAR

~~Choice: 2~~

~~GOT: 40~~

Q. Implement a Queue Using Linked List

31-Jan-2022

ALGORITHM

1. Declare a structure, Node with data and pointer to next node
2. Initialize FRONT, REAR as NULL

I. is Empty()

1. if (FRONT == REAR == NULL):
 return 1
2. return 0.

II. enqueue(ele)

1. Create new node, newNode with data as ele.
2. if (isEmpty()):
 FRONT = newNode
 else:
 REAR → next = newNode
3. newNode → next = NULL
4. REAR = newNode

III. dequeue()

1. temp_n = FRONT
2. temp = temp_n → data
3. FRONT = temp_n → next
4. if (FRONT == NULL):
 REAR = NULL.
5. Delete temp_n
6. return temp.

Choice: 2

Not: 40

Choice: 3

Queue Empty!!

Choice: 4,

N

ShowQueue()

1. if (isEmpty()):

DISPLAY "Queue Empty".

2. else:

P = FRONT

while (P ≠ NULL):

DISPLAY P → data

IV. main

1. Start

2. Display a menu with options Enqueue, Dequeue, Display and Exit.

3. INPUT choice

4. Based on choice, do:

if Enqueue:

INPUT element to enqueue, e

Call enqueue with e

if Dequeue:

if (isEmpty()):

DISPLAY "Queue Empty".

else:

Call dequeue, DISPLAY result

if Display:

Call ShowQueue()

if Exit:

End.

5. End

RESULT: Program is executed successfully and output is obtained.