**WEEK 5**

**AIM:** Queues and its various operations

In this exercise, I made a menu-driven program where I gave the user 5 choices. First 4 choices are enqueue, dequeue, search, display and the last choice is to exit the program.

I have implemented queue using linked list.

**VARIABLES USED**

Node of linked list: node

Data in node: data

Link to next node: next

Start of the queue: front

End of the queue: rear

Element to search: elem

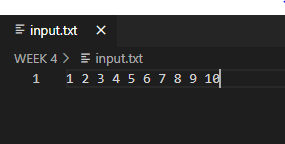
Element to push: data

Position of the number being searched: pos

Helper variable: choice,temp

File pointer: fptr

Initial content of input file:



All four operations are explained below in detail:

1. Enqueue: This function is used to insert data at the rear end of the queue.

Time complexity: O(1)

Sample Input from file: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Algorithm:

Step1. Get the value to be pushed in the queue in a varible data

Step2. Declare a temp variable of type pointer to node

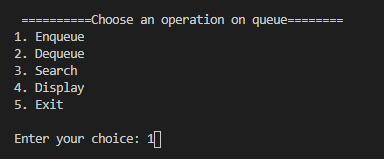
Step3. temp->data=data;

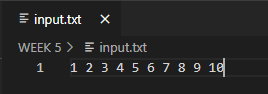
Step4. temp->next=NULL;

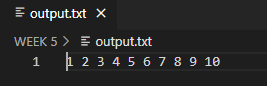
Step5. If rear is NULL then write front=rear=temp

Step6. Else rear->next=temp

Step7. rear=temp







1. Dequeue: This function is used to remove an element from the front end of the queue.

Time complexity: O(1)

Sample Output: 1 dequeued from the queue

Algorithm:

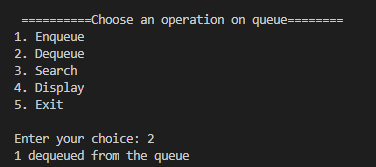
Step1. Declare a temp variable of type pointer to node and assign value of front to it.

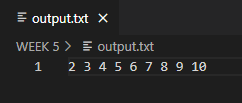
Step2. If temp is null then it is the underflow condition.

Step3. front=front->next

Step4. If front=NULL then make rear=NULL

Step5. free(temp)





1. Search: This function is used to search an element in queue.

Time complexity: O(n)

Sampe Input: 3

Sample Output: Element 3 found at position 2 in queue

Sample input: 1

Sample out: Element 1 not found

Algorithm:

Step1. Get the value to be searched in a varible elem

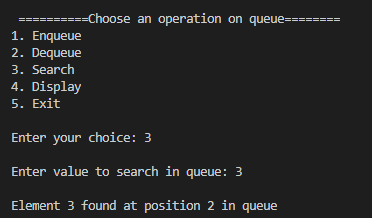
Step2. Declare a temp variable of type pointer to node and assign value of front to it.

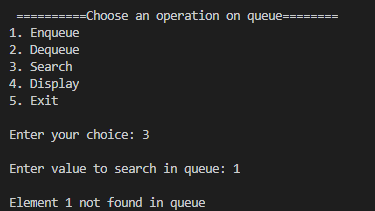
Step3. pos=0

Step4. Run a loop till the end of the queue and compare the data of the node with the value to be searched. Increment pos by one each time.

Step5. If they are equal, print the position and break the loop.

Step6. If temp is still null then number is not present.





1. Display: This function is used to print the content of the queue.

Sample Output: Queue: 2 3 4 5 6 7 8 9 10

Algorithm:

Step1. Declare a temp variable of type pointer to node and assign value of front to it.

Step2. If temp is null then the queue in empty.

Step3. Run a loop till the end of the queue and keep printing the data.

Step4. End the loop when you reach the end of linked list.

