

**Lab no.: 6 Date: February 13, 2024**

**Write a menu-based program to simulate the basic operations of Queue.**

A queue is a linear data structure where elements are stored in the **FIFO (First In First Out)** principle where the first element inserted would be the first element to be accessed. A queue is an **Abstract Data Type (ADT)** similar to stack, the thing that makes queue different from stack is that a queue is open at both its ends. The data is inserted into the queue through one end and deleted from it using the other end. Queue is very frequently used in most programming languages.

The basic stack operations are:

* Enqueue
* Dequeue

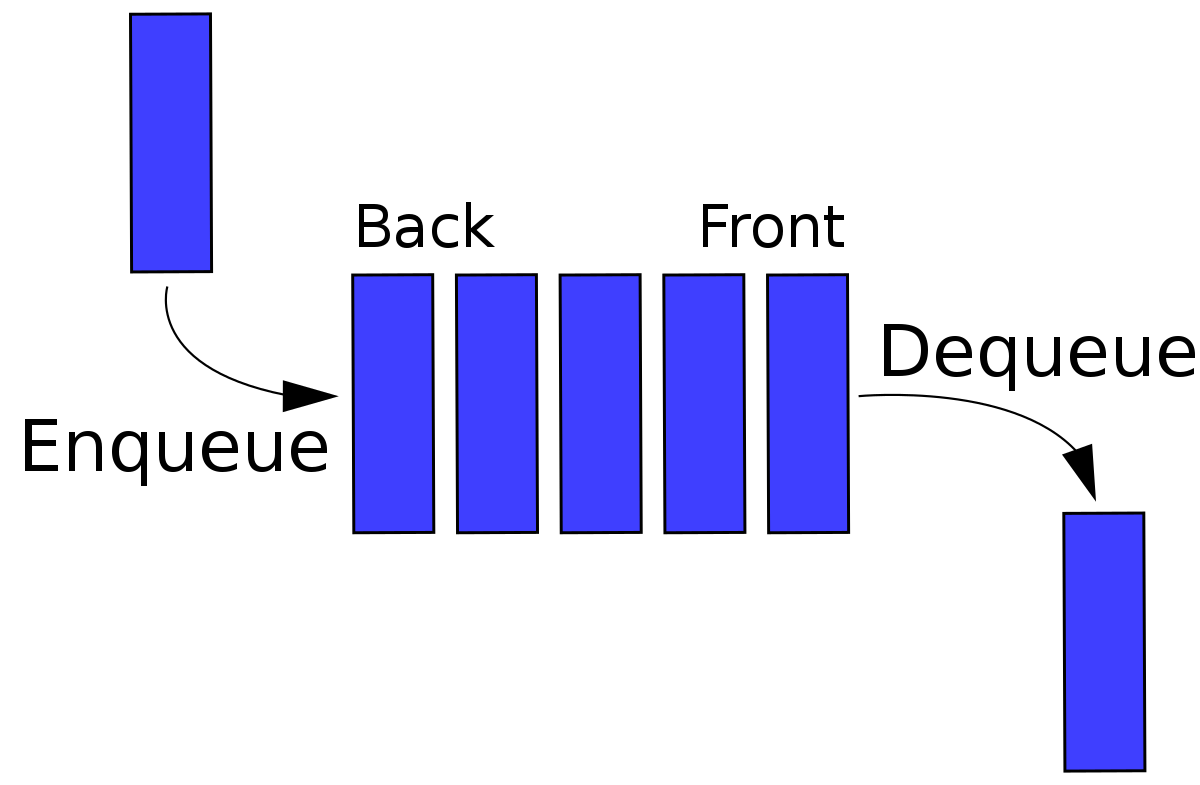


Figure 1: Queue Operation

**Programming Language: C**

**IDE: Microsoft Visual Code**

**Source code:**

/\* write a menu based program to simulate the basic operations of Queue\*/

#include <stdio.h>

#define MAXSIZE 15

int stack[MAXSIZE];

int front = -1;

int rear = -1;

int counter = -1;

// to enqueue data

void enqueue(int lrear)

{

if (front == -1 && rear == -1) // increasing the front value only once for the first time

front++;

if (lrear == MAXSIZE - 1)

printf("\nThe Queue is full.\n");

else

{

printf("Enter the data: ");

rear++;

scanf("%d", &stack[rear]);

}

}

// to dequeue data

int dequeue()

{

if (counter == MAXSIZE - 1)

printf("\nThe Queue has overflow (Front has reached to Rear).\nRestart the program to continue.\n");

else if (front == -1)

printf("\nThe Queue is empty.\n\n");

else

{

printf("%d", stack[front]);

counter++;

front++;

}

}

// to display data

void display(int lrear)

{

printf("\nThe data are:\n");

for (int i = front; i <= lrear; i++)

{

printf("%d ", stack[i]);

}

}

// main program

int main()

{

int choice;

while (1)

{

printf("\n\n....LINEAR QUEUE OPERATIONS...\n");

printf("1.Enqueue\n");

printf("2.Dequeue\n");

printf("3.Display\n");

printf("4.Exit\n");

printf("\nEnter you choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

enqueue(rear);

break;

case 2:

dequeue();

break;

case 3:

display(rear);

break;

case 4:

exit(0);

break;

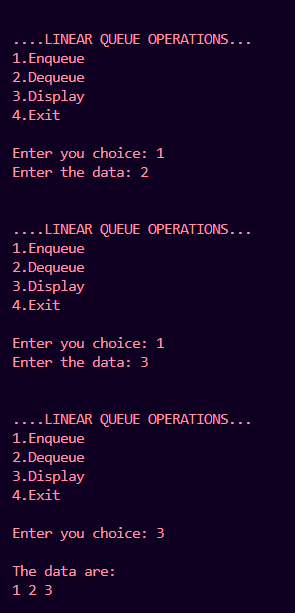
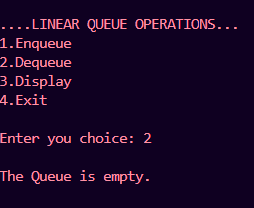
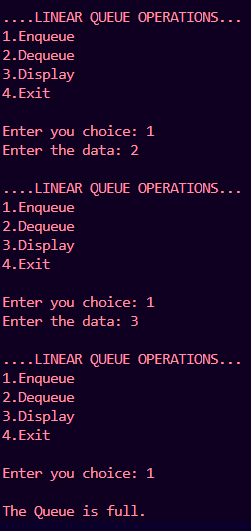
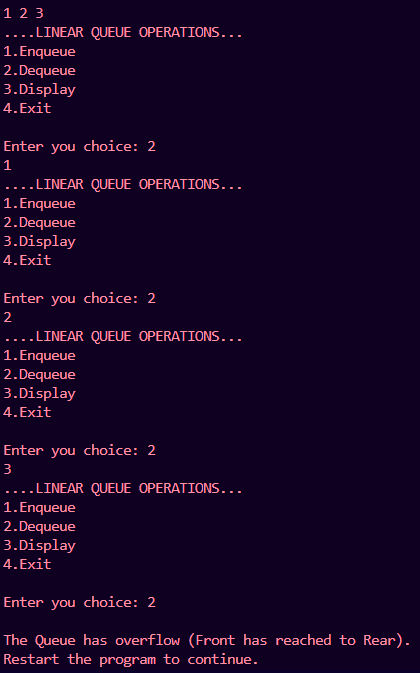
default:

printf("you didnt choose the right option.");

}

}

}



**Outputs:**