Harsh Patanwadia
69

PAGE No.	
DATE	/ /

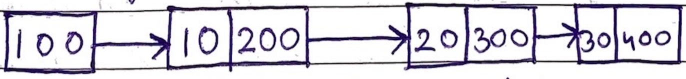
Expuriment No 2

Aim:- Implementation of queue using Array for real-world application.

Objective:- To introduce the concepts of data structure & analysis procedure.
To conceptualize linear data structure and its implementation for various real time applications

Theory

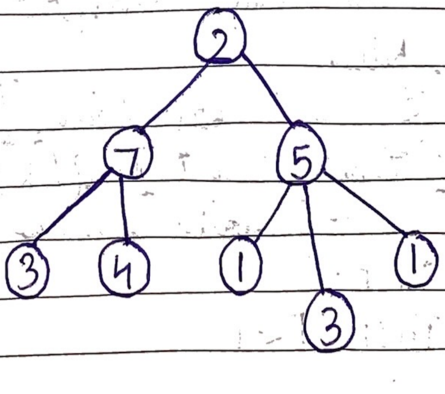
Introduction to linear & non-linear data structure
Linear data Structure - Organize data elements in linear fashion and each element is attached one after other.
- Continous memory locations allocation.



- Examples - Array, Stack, Queue, Lists

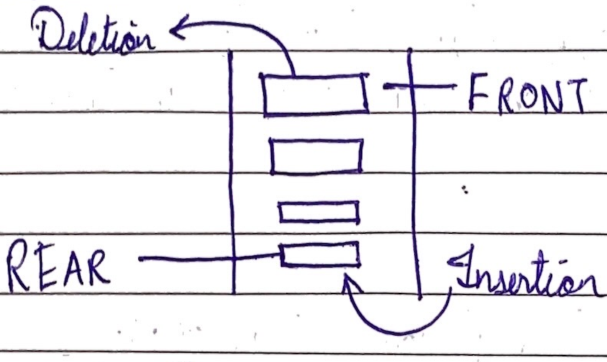
Non-Linear Data Structure - Organisation is not in a sequential fashion and its possible to attach a element to other several data elements multiple relationship among them.

- Examples - Graph, Tree



Introduction to Queue -

Queue is a linear structure which follows a particular order in which the operations are performed. The order is First in First out (FIFO). In a queue, new elements are added to queue from one end called REAR end & elements are always removed from other end called FRONT end.



Operations in Queue

Enqueue	-	Adds an item in queue
Dequeue	-	Removes an item in queue.
Front	=	get the front item from queue.
Rear		gets the rear item from queue.

Algorithm -

• Q INSERT (Q, F, R, N, Y). Given F & R pointers to front & rear elements of queue Q having N elements, elements Y insertion in queue Q

```
1  If R ≥ N
       then write ('Overflow')
       Return
2  [Increment rear pointer] R ← R+1
3  [Insert element] Q[R] ← Y
4  [Is front pointer properly set?]
   If F=0
   then F ← 1
   Return
```

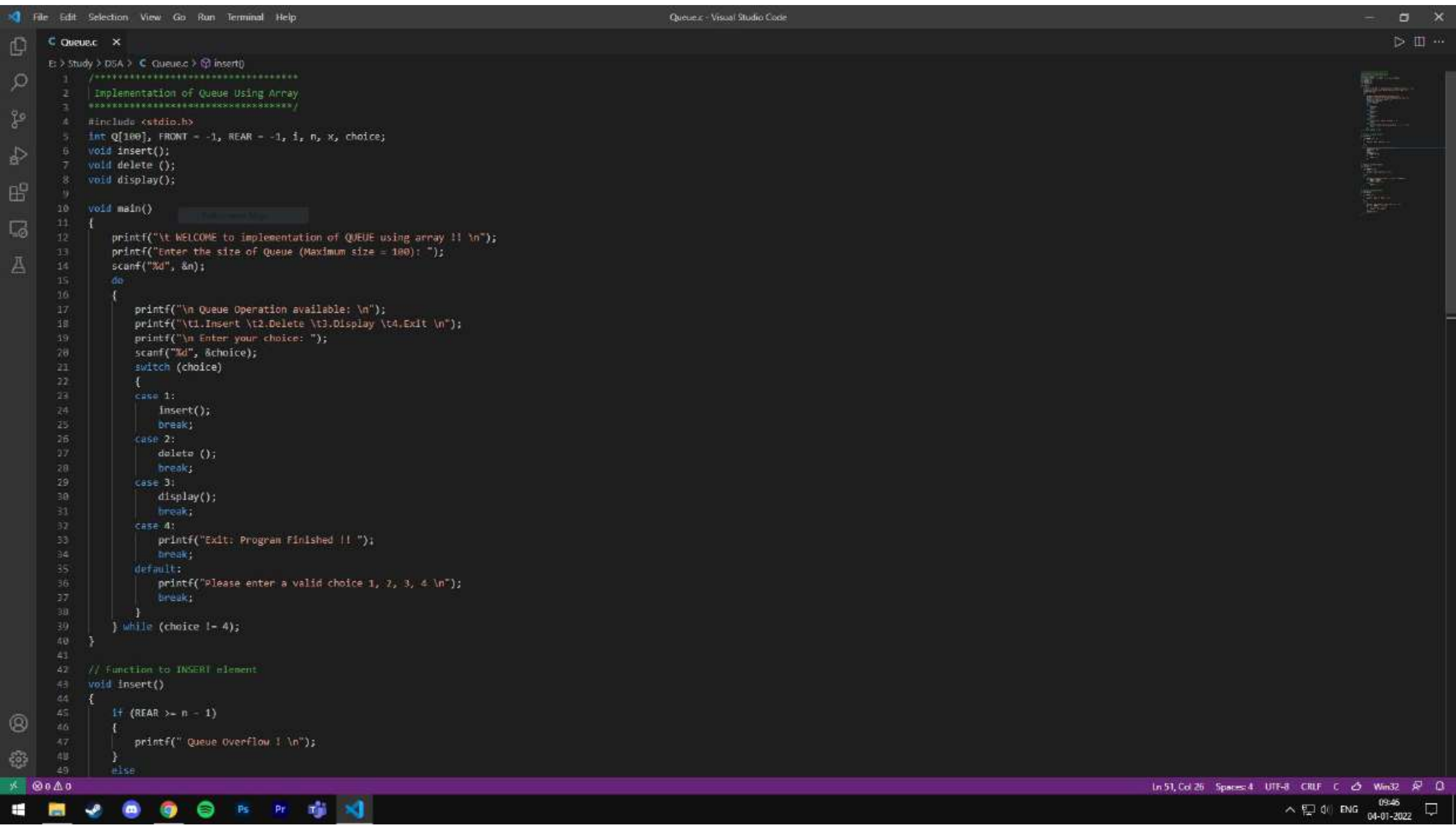
Q DELETE (Q, F, R), given F & R pointers to front of rear elements of queue Q, element Y is to be deleted

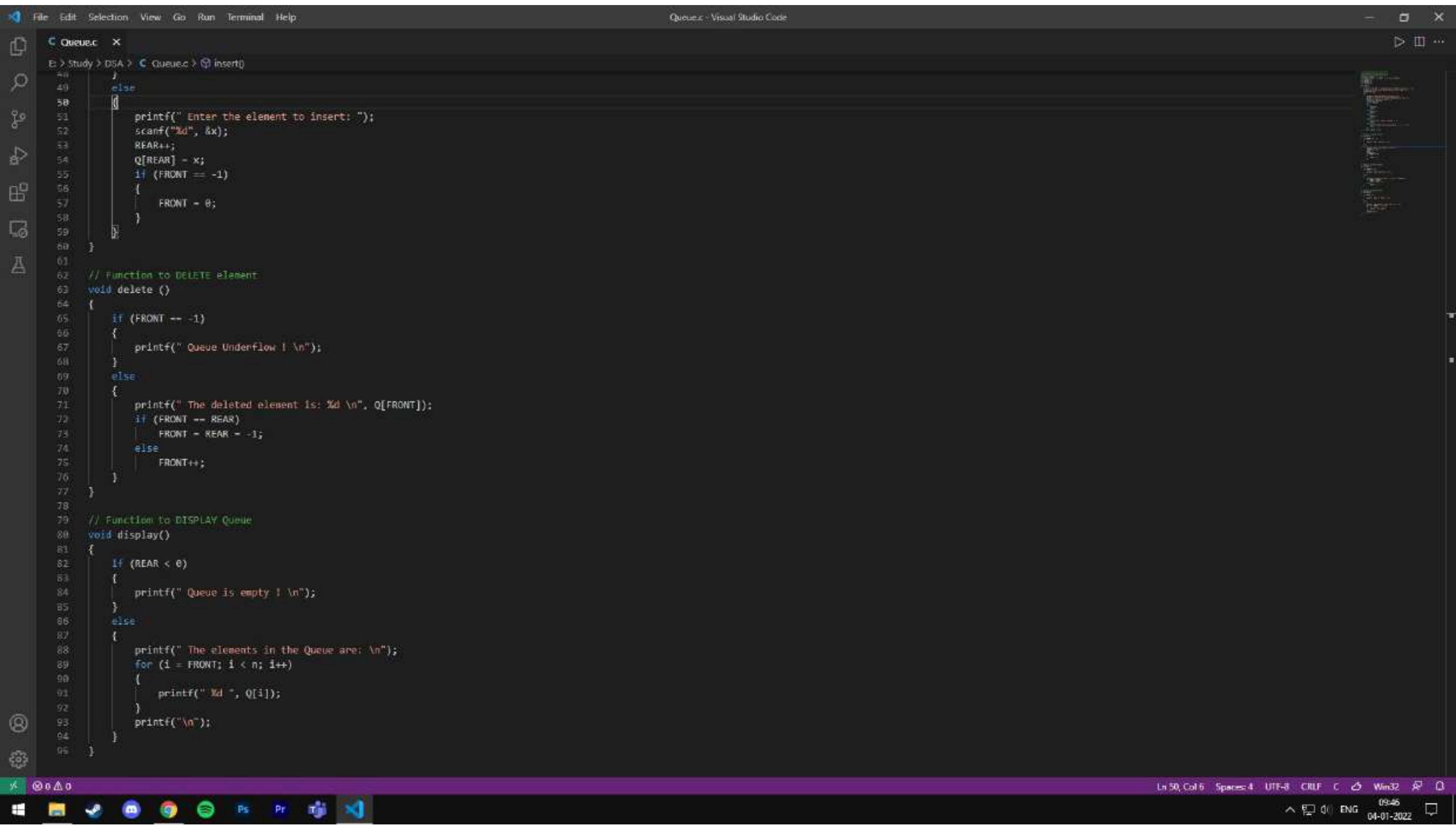
```
1  If F=0
     then write ('Underflow')
       Return (0)
2  [Delete element] Y ← Q[F]
3  [Delete empty]
   if F = R
   then F ← R ← 0
   else F ← F+1 (increment front pointer)
4  [Return element] Return [Y]
```

Example:- People standing in a railway reservation row for tickets. As such new person comes and stands at end of row and person after their reservation confirmation get out of row from front end.

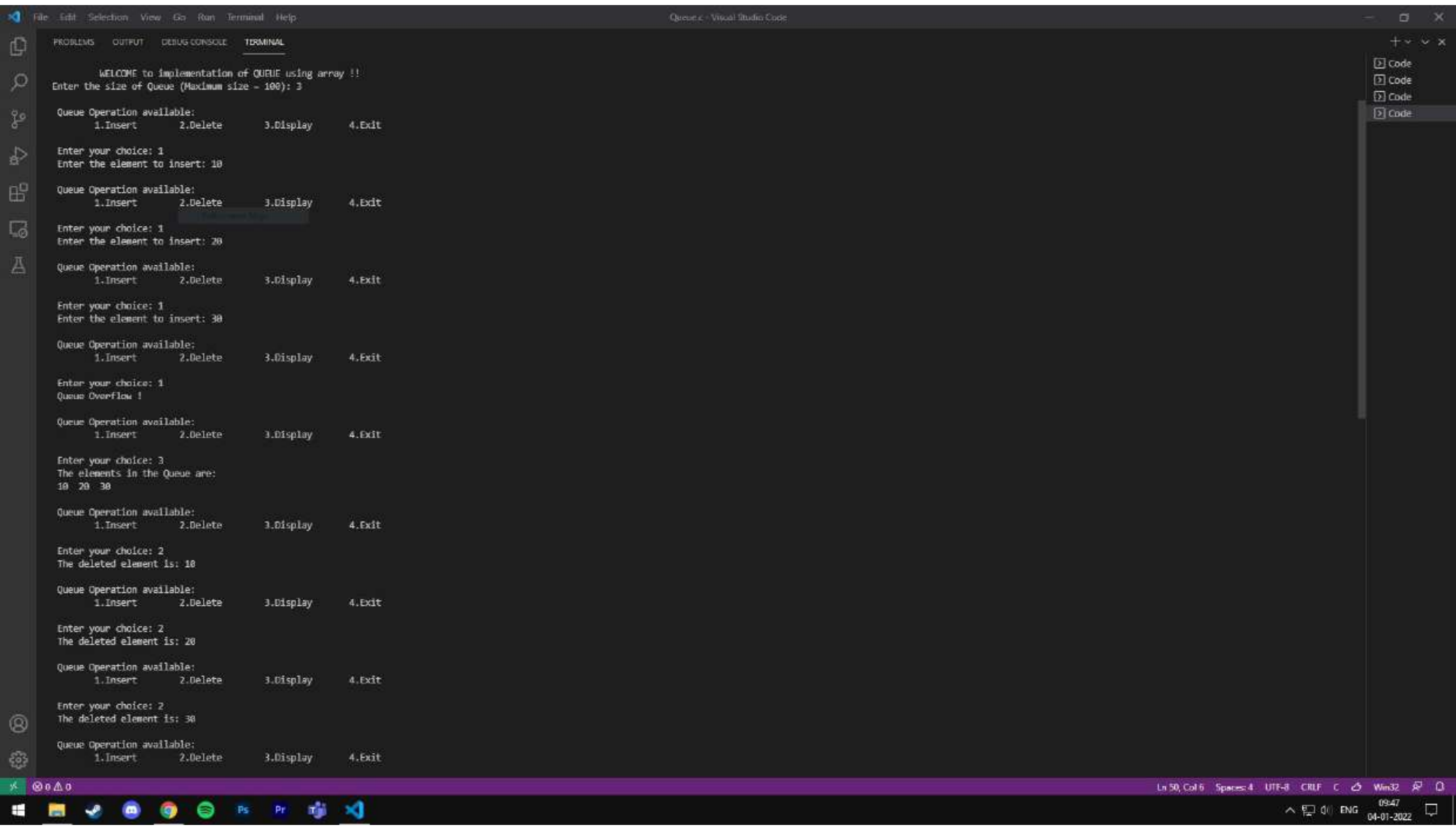
Conclusion:- Learned how to implement queue using array. and Queue is used when things don't have to be processed imediately but have to processed in first in first out.

Outcome:- Apply the concepts of queue for real-world application.

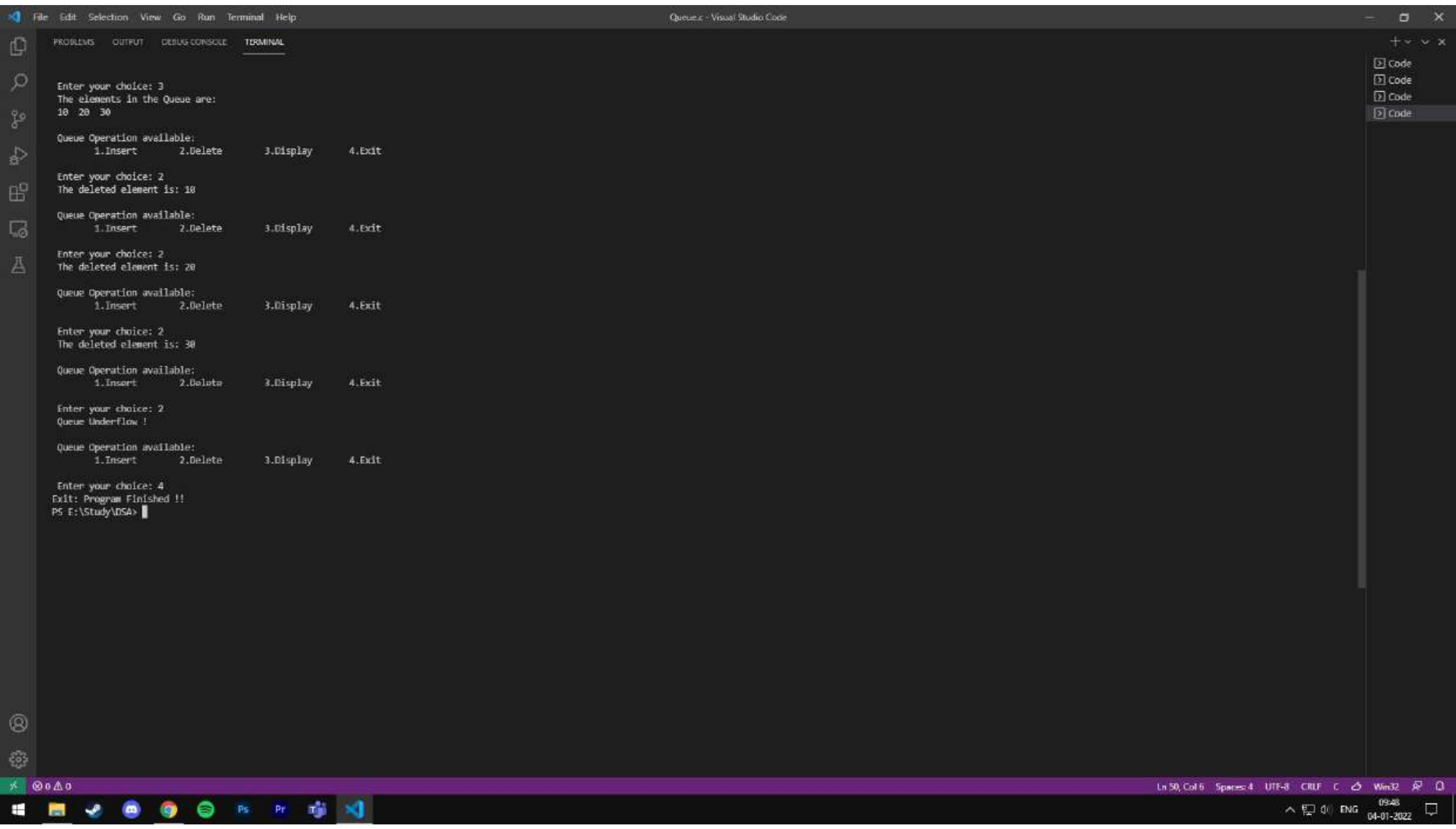

```c
/***********************************
 Implementation of Queue Using Array
***********************************/
#include <stdio.h>
int Q[100], FRONT = -1, REAR = -1, i, n, x, choice;
void insert();
void delete ();
void display();

void main()
{
    printf("\t WELCOME to implementation of QUEUE using array !! \n");
    printf("Enter the size of Queue (Maximum size = 100): ");
    scanf("%d", &n);
    do
    {
        printf("\n Queue Operation available: \n");
        printf("\t1.Insert \t2.Delete \t3.Display \t4.Exit \n");
        printf("\n Enter your choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            insert();
            break;
        case 2:
            delete ();
            break;
        case 3:
            display();
            break;
        case 4:
            printf("Exit: Program Finished !! ");
            break;
        default:
            printf("Please enter a valid choice 1, 2, 3, 4 \n");
            break;
        }
    } while (choice != 4);
}

// Function to INSERT element
void insert()
{
    if (REAR >= n - 1)
    {
        printf(" Queue Overflow ! \n");
    }
    else
```


File Edit Selection View Go Run Terminal Help
Queue.c - Visual Studio Code
Queue.c
E: > Study > DSA > Queue.c > insert()
Ln 50, Col 6 Spaces: 4 UTF-8 CRLF C Win32
ENG
09:46
04-01-2022

```c
    }
    else
    {
        printf(" Enter the element to insert: ");
        scanf("%d", &x);
        REAR++;
        Q[REAR] = x;
        if (FRONT == -1)
        {
            FRONT = 0;
        }
    }
}

// Function to DELETE element
void delete ()
{
    if (FRONT == -1)
    {
        printf(" Queue Underflow ! \n");
    }
    else
    {
        printf(" The deleted element is: %d \n", Q[FRONT]);
        if (FRONT == REAR)
            FRONT = REAR = -1;
        else
            FRONT++;
    }
}

// Function to DISPLAY Queue
void display()
{
    if (REAR < 0)
    {
        printf(" Queue is empty ! \n");
    }
    else
    {
        printf(" The elements in the Queue are: \n");
        for (i = FRONT; i < n; i++)
        {
            printf(" %d ", Q[i]);
        }
        printf("\n");
    }
}
```


File Edit Selection View Go Run Terminal Help
Queue.c - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Code
Code
Code
Ln 50, Col 6 Spaces: 4 UTF-8 CRLF C Win32
ENG
09:47
04-01-2022

```
        WELCOME to implementation of QUEUE using array !!
Enter the size of Queue (Maximum size - 100): 3

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 1
 Enter the element to insert: 10

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 1
 Enter the element to insert: 20

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 1
 Enter the element to insert: 30

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 1
 Queue Overflow !

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 3
 The elements in the Queue are:
 10  20  30

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 2
 The deleted element is: 10

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 2
 The deleted element is: 20

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 2
 The deleted element is: 30

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit
```


File Edit Selection View Go Run Terminal Help
Queue.c - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Code
Code
Code
Ln 50, Col 6 Spaces: 4 UTF-8 CRLF C Win32
ENG
09:48
04-01-2022

```
 Enter your choice: 3
 The elements in the Queue are:
 10  20  30

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 2
 The deleted element is: 10

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 2
 The deleted element is: 20

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 2
 The deleted element is: 30

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 2
 Queue Underflow !

 Queue Operation available:
        1.Insert        2.Delete        3.Display       4.Exit

 Enter your choice: 4
Exit: Program Finished !!
PS E:\Study\DSA>
```