

15/10/24 Week-03

define MAX 3

declare rear = -1

declare front = -1

declare queue [max-size]

function isEmpty()

return (front == -1 and rear == -1)

function isFull()

return (rear == MAX - 1)

function Insert (value)

If isFull () Then

Print Queue is full

Else

If (front == -1)

front = 0;

Print Element to be inserted

Scan the Element

rear = rear + 1

queue [rear] = 1

function Delete

If isEmpty

Print Underflow

return

Else

Print Element to be inserted

front = front + 1

function display ()

if (front == -1)

Print (Queue is Empty)

else

Print (Queue)

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

print (queue[i])

End.

Program

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define MAX 3
```

```
int t, ch, queue[MAX], rear = 1, front = -1, item;
```

```
void insert();
```

```
void delete();
```

```
void display();
```

```
int main () {
```

```
    while (1) {
```

printf ("Enter the choice\n");

printf ("1 : Insertion 2 : deletion 3 : display 4 : Exit ");

```
    scanf ("%d", &ch);
```

```
    switch (ch) {
```

Case 1 : insert();

```
            break;
```

Case 2 : delete ();

```
            break;
```

Case 3 : display ();

```
            break;
```

Case 4 : EXIT (0);

default :

printf ("Invalid choice");

```
} }
```



```
void insert () {
    if (rear == MAX - 1) {
        if (front == 0) {
            printf("Overflow!\n");
            return;
        } else {
            front = 0;
            rear = -1;
        }
    }
    printf("The element need to be inserted ");
    scanf("%d", &item);
    rear++;
    queue[rear] = item;
    if (front == -1) {
        front = 0;
    }
}

void delete () {
    if (front == -1 || front > rear) {
        printf("Queue is underflow\n");
        return;
    } else {
        printf("The element deleted is : %d\n", queue[front]);
        front++;
    }
}

if (front > rear) {
    front = rear = -1;
}
```

```

void display() {
    if (front == -1 || front > rear) {
        printf("Queue is empty\n");
    } else {
        printf("The elements in the queue are:\n");
        for (int i = front; i <= rear; i++) {
            printf("%d\n", queue[i]);
        }
    }
}

```

Output:

Perform Queue Operation:

1. Insert
2. delete
3. display
4. Exit

Enter your choice : 1

Enter the element to be added: 1

Perform queue operation:

1. Insert
2. delete
3. display
4. Exit

The elements in the queue are: 10, 20, 30.

Perform queue operation:

1. Insert
2. delete
3. display
4. Exit

Enter your choice : 2

The deleted element is 10.

~~Deleted
Element
is 10~~



leet 314. ClearDigits

```
class Solution {
    string clearDigits(string s) {
        int l = s.length();
        for (int i = 0; i < l; i++) {
            if (isDigit(s[i])) {
                if (s[i] == '0') {s.erase(i-1, 2); i+=2; i++; }
                else {s.erase(i, 2); i-=3; i++; }
            }
        }
    }
}
```

Case 1

Input s: "ab0"
Output "abc"
Expected "abc"

Case 2

s = "0b34"
output "
Expected "

~~Richard P
15/10/24~~

o. 30.

ent u. 10.