

Name – Bhakti Bapurao Patil

Reg.no – 2020BIT064

Assignment NO 1 – Write a program to perform stack and queue

Stack

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
#define MAX 1000
```

```
class Stack {
```

```
    int top;
```

```
public:
```

```
    int a[MAX];
```

```
    Stack() { top = -1; }
```

```
    bool push(int x);
```

```
    int pop();
```

```
    int peek();
```

```
    bool isEmpty();
```

```
};
```

```
bool Stack::push(int x)
```

```
{
    if (top >= (MAX - 1)) {
        cout << "Stack Overflow";
        return false;
    }
    else {
        a[++top] = x;
        cout << x << " pushed into stack\n";
        return true;
    }
}
```

```
int Stack::pop()
{
    if (top < 0) {
        cout << "Stack Underflow";
        return 0;
    }
    else {
        int x = a[top--];
        return x;
    }
}
```

```
int Stack::peek()
{
```

```

        if (top < 0) {
            cout << "Stack is Empty";
            return 0;
        }
        else {
            int x = a[top];
            return x;
        }
    }
}

```

```

bool Stack::isEmpty()
{
    return (top < 0);
}

```

```

int main()
{
    class Stack s;
    s.push(10);
    s.push(20);
    s.push(30);
    cout << s.pop() << " Popped from stack\n";
    cout << "Top element is : " << s.peek() << endl;

    cout << "Elements present in stack : ";
}

```

```

        while(!s.isEmpty())
        {
            cout << s.peek() <<" ";
            s.pop();
        }

        return 0;
    }
}

```

The screenshot shows the Programiz C++ Online Compiler interface. The code editor on the left contains a C++ program for a stack. The output window on the right shows the results of running the program.

```

main.cpp
1 // Online C++ compiler to run C++ program online
2 /* C++ program to implement basic stack
3 operations */
4 #include <bits/stdc++.h>
5
6 using namespace std;
7
8 #define MAX 1000
9
10 class Stack {
11     int top;
12
13 public:
14     int a[MAX]; // Maximum size of Stack
15
16     Stack() { top = -1; }
17     bool push(int x);
18     int pop();
19     int peek();
20     bool isEmpty();
21 };
22
23 bool Stack::push(int x)

```

Output:

```

/tmp/7m34d52958.o
10 pushed into stack
20 pushed into stack
30 pushed into stack
30 Popped from stack
Top element is : 20
Elements present in stack : 20 10

```

2) queue

```

#include <bits/stdc++.h>

using namespace std;

class Queue {
public:
    int front, rear, size;

```

```

        unsigned capacity;
        int* array;
    };

Queue* createQueue(unsigned capacity)
{
    Queue* queue = new Queue();
    queue->capacity = capacity;
    queue->front = queue->size = 0;

    queue->rear = capacity - 1;
    queue->array = new int[queue->capacity];
    return queue;
}

int isFull(Queue* queue)
{
    return (queue->size == queue->capacity);
}

int isEmpty(Queue* queue)
{
    return (queue->size == 0);
}

void enqueue(Queue* queue, int item)
{
    if (isFull(queue))

```

```

        return;

queue->rear = (queue->rear + 1)
                % queue->capacity;

queue->array[queue->rear] = item;

queue->size = queue->size + 1;

cout << item << " enqueued to queue\n";

}

int dequeue(Queue* queue)
{
    if (isEmpty(queue))
        return INT_MIN;

    int item = queue->array[queue->front];

    queue->front = (queue->front + 1)
                    % queue->capacity;

    queue->size = queue->size - 1;

    return item;
}

int front(Queue* queue)
{
    if (isEmpty(queue))
        return INT_MIN;

    return queue->array[queue->front];
}

int rear(Queue* queue)
{

```

```
        if (isEmpty(queue))
            return INT_MIN;
        return queue->array[queue->rear];
    }
int main()
{
    Queue* queue = createQueue(1000);

    enqueue(queue, 10);
    enqueue(queue, 20);
    enqueue(queue, 30);
    enqueue(queue, 40);

    cout << dequeue(queue)
          << " dequeued from queue\n";

    cout << "Front item is "
          << front(queue) << endl;
    cout << "Rear item is "
          << rear(queue) << endl;

    return 0;
}
```

Practice Problems | C++ | Little Chef and Sums | Online C++ Compiler | Stack and Queue C++ | Introduction and Array | (T) WhatsApp

programiz.com/cpp-programming/online-compiler/

Programiz C++ Online Compiler Interactive C++ Course

main.cpp

```
1 // CPP program for array
2 // implementation of queue
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 // A structure to represent a queue
7 class Queue {
8 public:
9     int front, rear, size;
10    unsigned capacity;
11    int* array;
12 };
13
14 // function to create a queue
15 // of given capacity.
16 // It initializes size of queue as 0
17 Queue* createQueue(unsigned capacity)
18 {
19     Queue* queue = new Queue();
20     queue->capacity = capacity;
21     queue->front = queue->size = 0;
22
23     // This is to implement the queue
```

Output

```
//tmp/7n34d5295b.o
10 enqueued to queue
20 enqueued to queue
30 enqueued to queue
40 enqueued to queue
10 dequeued from queue
Front item is 20
Rear item is 40
```

Type here to search 20°C Clear 6:30 28-01-2023