```
#pragma once
#include <iostream>

#define CAPACITY 10

class STACK_FULL :std::exception
{

};

class STACK_EMPTY : std::exception
{

};

template <typename T>
class FixSizeStack
{
        T arr[CAPACITY];
        int top;

public:

        FixSizeStack()
        {
                this->top = -1;
        }

        bool isFull()
        {
                return top == (CAPACITY-1);
        }

        bool isEmpty()
        {
                return top == -1;
        }

        void push(T data)
        {
                if (isFull())
                {
                        throw new STACK_FULL;
                }
                else
                {
                        this->arr[++top] = data;
                }
        }

        T pop()
        {
                if (isEmpty())
                {
                        throw new STACK_EMPTY;
                }
```

```
                else
                {
                        return this->arr[top--];

                }
        }

        T topVal()
        {
                if (isEmpty())
                {
                        throw new STACK_EMPTY;
                }
                else
                {
                        return this->arr[top];
                }
        }

};
```