Pila Statica:

```
#ifndef STATIC_STACK_H
#define STATIC_STACK_H

template <typename T>
class Stack_static
{
 T * vet;
 int top = -1;
 int max_size;
 int size = 0;
public:
 Stack_static (int max_size = 10): max_size (max_size) {vet = new T [max_size];}
 bool isEmpty() {return size == 0;}
 bool isFull() {return size == max_size;}
 T getTop() {
  if (isEmpty())
   throw "Void Stack!";
  return vet [top]
 }
 void push (T val)
  if (isFull())
  {
   cerr << "Full stack" << endl;
   return;
  }
  vet [++top] = val;
  size++;
  return;
}
```

```
T getLast()
{
 if (isEmpty())
  throw "Empty Stack";
 return vet[top];
}
```

```
T pop()
{
 if (isEmpty())
  throw "Void Stack";
 size--;
 return vet [top--];
}
```

Operazioni in pila Statica:

Push:

```
void push (T val) // inserimento di un valore val
  if (isFull()) // Controllo se è piena (se lo è non potrò inserire altri elementi)
  {
    cerr << "Full stack" << endl;
    return;
  }
  vet[++top] = val; // Preincremento top e assegno val in quella posizione
  size++;           // incremento size => il numero di elementi nella
  return; // esco dalla funzione
}
```

```
T pop()
{
  if (isEmpty())
    throw "Void Stack";
  size--; // decremento perché ho tolto un elemento.
  return vet[top--]; // ritorniamo la posizione del top decrementato.
}
```

Pila Dinamica: