

Stack

pila vuota

push (5)

5

push (7)

7
5

push (-3)

-3
7
5

push (1)

1
-3
7
5

il metodo pop() non accetta argomenti e rimuove il primo elemento della pila (o l'ultimo inserito) e lo ritorna come risultato

Accesso nella Stack: LIFO

```
#include <iostream>
```

```
using namespace std;
```

```
class Stack {
```

```
private:
```

```
int* A;
```

```
int top;
```

```
int dim;
```

```
void allarga() {
```

```
int* tmp = new int [dim + 40]; //creo un array nuovo temporaneo più grande
```

```
for (int i=0; i < dim; i++) { //ci copio l'array A
```

```
tmp[i] = A[i];
```

```
}
```

```
delete[] A; //cancello l'array appena copiato
```

```
this->A = tmp; //copio l'array temporaneo in A
```

```
dim = dim + 40; //aggiorno la dimensione di A
```

```
}
```

```
public:
```

```
Stack() { //crea la pila vuota
```

```
A = NULL;
```



```

top = 0;

dim = 0;

}

void push (int x) {

    if (top == dim) {

        this->allarga();

    }

    A[top] = x;

    top++;

}

int pop() {

    if (isEmpty()) {

        throw out_of_range ("Pila vuota");

    }

    top--;

    return A[top];

}

bool isEmpty() const {

    return top == 0;

}

int size() const {

    return top;

}

int getElement (int i) const {

```



```

        return AC13;

    }

};

ostream& operator<< (ostream & dest, const Stock& s) {

    dest << "[ ";

    for (int i = 0; i < s.GetSize(); i++) {

        dest << s.GetElement(i);

    }

    if ((i < s.GetSize() - 1)) {

        dest << ", ";

    }

}

dest << " ]";

return dest;

}

int main () {

    return 0;

}

```