

Esercizio 1

Implementare un algoritmo che data una coda di interi ne costruisca un'altra ottenuta dalla prima considerando solo i valori positivi.

Esercizio 2

Progettare ed implementare un calsse `MultipleStack` che fornisca la gestione di m stack. La dichiarazione della classe dovrebbe essere simile a qualcosa del genere:

```
class MultipleStack
{
    // ...
public:
    MultipleStack (unsigned int);
    void Push (Object&, unsigned int);
    Object& Pop (unsigned int);
    // ...
};
```

- Il costruttore ha in input un argomento di tipo intero che specifica il numero di stack da gestire
- la funzione `Push` ha due argomenti. Il primo è l'oggetto da inserire e il secondo specifica lo stack in cui inserirlo
- la funzione `Pop` ha in input un solo intero che specifica lo stack da cui eliminare un oggetto

Scegliere una delle seguenti implementazioni:

- tenere tutti gli stack in un solo array
- usare un array di oggetti di tipo `Stack`
- Usare una lista collegata di oggetti di tipo `Stack`

Esercizio 3

Implementare un algoritmo che data una coda di interi ne costruisca un'altra ottenuta dalla prima considerando solo i valori positivi e conservando la coda originaria.

Esercizio 4

Il tipo di dato **deque** (double ended queue) è una sequenza modificabile ad entrambi gli estremi, in cui è possibile inserire un emlemento in testa o inserirlo in fondo, e cancellare un elemento dalla testa o cancellarlo dal fondo. Si fornisca una realizzazione in C++ (operatori con tempo $O(1)$).

Esercizio 5

Pila senza duplicati modificare l'operatore `inPila` (con tempo $O(n)$) per una Pila che non ammette elementi duplicati.

```
L      L
A      L A
*  A   L
S      L S
T      L S T
I              L S T I
*  I   L S T
N      L S T N
S      L S T N      (S già presente)
*  N   L S T
L      L S T
I      L S T I      (I già presente)
```

Esercizio 6

Implementare un interprete per un linguaggio di programmazione, in cui ogni programma consista di una singola operazione aritmetica, preceduta da una sequenza di istruzioni di assegnamento in cui possono comparire espressioni aritmetiche che coinvolgono interi e variabili in cui nome è dato da una singola lettera minuscola. Per esempio dato l'ingresso

```
(x = 1)
(y = (x + 1))
(((x + y) * 3) + (4 * x))
```

il programma dovrebbe dare in uscita il valore 13.

Esercizio 7

Convertire un'espressione aritmetica da notazione infissa alla corrispondente notazione postfissa utilizzando una coda come risultato della conversione.

espressione	pila	coda
35 + 17 * (40 - 9) - 7		
+ 17 * (40 - 9) - 7		35
17 * (40 - 9) - 7	+	35
* (40 - 9) - 7	+	17 35
(40 - 9) - 7	+ *	17 35
40 - 9) - 7	+ * (17 35
- 9) - 7	+ * (40 17 35
9) - 7	+ * (-	40 17 35
) - 7	+ * (-	9 40 17 35
- 7	+ * (- 9 40 17 35
7	-	+ * - 9 40 17 35
	-	7 + * - 9 40 17 35
		- 7 + * - 9 40 17 35