

## Descrizione del problema

Si vuole progettare un algoritmo per risolvere il seguente problema. Data una sequenza di interi, verificare se la sequenza è palindroma, ovvero è la stessa sequenza se viene letta da sinistra verso destra o da destra verso sinistra.

Ad esempio l'istanza [1, 2, 3, 2, 1] è positiva; l'istanza [1, 2, 3, 3, 2, 1] è positiva; l'istanza [1, 2, 3, 1, 2, 3] è negativa.

## Task

**(1 pt)** Descrivere in modo sintetico la **specificità del problema**.

**(1 pt)** Indicare **di che tipo di problema si tratta** (accumulazione, conteggio, verifica esistenziale, verifica universale, ricerca, minimo/massimo).

**(3.5 pt)** Descrivere un algoritmo risolutivo per il problema utilizzando un **diagramma a blocchi**.

Scrivere un **programma C Palindroma** in cui sono definiti:

**(6 pt)** Una **funzione palindroma** con parametri: 1) un riferimento ad un array di interi; e 2) un intero che rappresenta la lunghezza dell'array. La funzione restituisce 1 se la sequenza di interi è palindroma, 0 altrimenti.

**(2.5 pt)** Una **funzione main** che gestisce l'interazione con l'utente. La funzione main deve:

- chiedere all'utente quanti interi ha la sequenza e leggere la risposta dell'utente
- chiedere all'utente di introdurre gli interi della sequenza, leggere gli interi introdotti dall'utente e memorizzarli in un array
- invocare la funzione palindroma, fornendogli come parametro un riferimento alla sequenza di interi appena letta e la sua lunghezza
- stampare un messaggio che comunica all'utente se la sequenza è palindroma oppure no

## Come e cosa consegnare

Consegnare un unico file **main.c** che, oltre al programma, contiene la specificità ed il tipo di problema (il diagramma a blocchi che illustra l'algoritmo risolutivo può essere disegnato su un foglio a parte), commentati come nell'esempio che segue.

```
/* SPECIFICA
```

```
* Input: ....
```

```
* Pre-condizione: ...
```

```
* Output: ....
```

```
* Post-condizione: ...
```

```
* TIPO DI PROBLEMA: ...
```

```
*/
```

```
#include <stdio.h>
```

```
...
```