

# Laboratorio di programmazione e Informatica 1

## - A.A. 2019-2020 -

Prova di laboratorio: Quarto appello - 22 settembre 2020

**ISTRUZIONI:** La prima riga di ogni programma C deve contenere il proprio nome e cognome. I programmi devono essere strutturati in funzioni e completi di commenti che spieghino il procedimento.

**ATTENZIONE!** Non saranno valutati programmi che non passano la fase di compilazione. Si consiglia pertanto di “mettere sotto commento” le parti di programma che danno errore in compilazione.

**Tempo a disposizione: 3 ore.**

## ESERCIZI

### Esercizio 1

**Def1** Un indice  $k$  di un vettore di interi  $a$  lungo  $n$  ( $0 \leq k < n$ ) è il *baricentro* di  $a$ , se la somma degli elementi di  $a$  prima di  $k$  è uguale alla somma degli elementi a cominciare da  $k$  fino alla fine. Formalmente:

$$\sum_{i=0}^{k-1} a[i] = \sum_{i=k}^{n-1} a[i]$$

**Def2** Una rotazione di ordine  $r$  di un vettore di interi  $a$  lungo  $n$  ( $0 \leq r < n$ ) è un nuovo vettore  $b$  ottenuto spostando ordinatamente gli elementi  $a[0], a[1], \dots, a[r-1]$  alla fine del vettore. Formalmente

$$b = a[r] \dots a[n-1] a[0] a[1] \dots a[r-1]$$

- Si scriva una funzione C `int baricentro(int v[], int n, int *k)` che restituisce 1 se esiste almeno un baricentro nel vettore  $v$  e 0 altrimenti. Quando torna 1, essa carica nel parametro  $k$  l'indice del baricentro.
- Scrivere un programma in C che:
  1. Chiede all'utente gli estremi di un intervallo di valori interi da cui estrarre dei valori random. (L'intervallo può contenere valori sia positivi che negativi).
  2. Genera un vettore di dimensione 20 contenente valori random presi dall'intervallo prescelto.
  3. Stampa il vettore sullo schermo.
  4. Utilizza la funzione `baricentro` per verificare se il vettore stesso o una sua rotazione ammette un baricentro.

5. Stampa sullo schermo se ha trovato una rotazione del vettore che ammette baricentro e, in caso affermativo, stampa tale rotazione e il valore del baricentro.
6. Discutere della complessità dell'algoritmo proposto. Nell'ipotesi che il vettore contenga valori tutti positivi, si può dare un algoritmo con complessità diversa?

## Esercizio 2

Scrivere un programma in linguaggio C che legge dall'utente una serie di stringhe. L'immissione termina quando l'utente inserisce `zzz`. A seguito di ogni immissione si crea e si stampa sullo schermo la stringa palindroma ottenuta concatenando la stringa immessa dall'utente con la sua "reverse".

Ad esempio: se l'utente ha immesso `programma`, dovrà essere generata e stampata la stringa `programmaammargorp`. Non è consentito utilizzare funzioni di C per la gestione stringhe (utilizzare solo confronti tra caratteri).

Si assuma che l'utente inserisca stringhe di lunghezza inferiore a 20.

## Esercizio 3

Una relazione simmetrica definita in un insieme di  $n$  oggetti numerati da 1 a  $n$  può essere descritta da un insieme finito di coppie  $(a, b)$  con  $a, b \in \{1, \dots, n\}$  tale che la presenza della coppia  $(a, b)$  implica la presenza della coppia  $(b, a)$ . Si assuma tale relazione (insieme di coppie) descritta con una lista. Ogni nodo contiene due campi per valori interi e un campo puntatore a tale nodo lista.

Scrivere una funzione in C che, data una lista di coppie interi positivi, verifichi se questa rappresenta *correttamente* una relazione simmetrica. In particolare la funzione controlla se la presenza della coppia  $(a, b)$  implica la presenza della coppia  $(b, a)$ . La funzione dovrà inoltre modificare la lista aggiungendo gli eventuali nodi (senza utilizzare funzioni aggiuntive) per soddisfare la condizione di simmetria. La funzione restituirà 1 se la lista non è stata modificata o 0 se è stata modificata.

VERSIONE SEMPLIFICATA: la funzione controlla soltanto se la relazione è simmetrica e restituisce 1 se è simmetrica e 0 altrimenti (senza inserire nodi).

Inserire tali funzioni nel file `ListaBase3App.c` allegato.