Laboratorio di programmazione e Informatica 1
- A.A. 2024-2025 -

Sesto-appello - 16 settembre 2025

ISTRUZIONI:

- Gli esercizi vanno chiamati come CognomeNomeEse1.c, CognomeNomeEse2.c,CognomeNomeEse3.c e consegnati tramite la chiavetta Usb del docente. (Consegnare soltanto i file .c)
- La prima riga di ogni programma C deve contenere il proprio nome e cognome come commento.
- I programmi devono essere strutturati in funzioni e completi di commenti che spieghino il procedimento.
- Quando è richiesto lo studio della complessità dell'algoritmo implementato, questa va descritta alla fine del programma, sempre sotto commento.
- Ogni esercizio vale 10 punti. Bisogna fare ≥ 18 punti con almeno 5 punti su ogni esercizio.
- **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.

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

Siano $P_0, P_1, \ldots, P_{n-1}$ punti nel piano. Alcuni di questi punti sono uniti da segmenti; un *triangolo* è costituito da tre punti tutti uniti da segmenti. I segmenti tra i punti sono descritti da una matrice S, booleana (cioè a valori 0 e 1) e quadrata di lato n come segue: i punti P_x e P_y sono uniti da un segmento se e solo se nella matrice $S[x][y] = 1$. Ad esempio la seguente matrice:

	0	1	2	3	4
0	1	1	1	0	1
1	1	1	0	0	1
2	1	0	1	0	1
3	0	0	0	1	0
4	1	1	1	0	1

descrive i segmenti tra i cinque punti $P_0, ..., P_4$. Possiamo osservare che i punti P_0, P_2, P_4 costituiscono un triangolo ($S[0][2] = 1$, $S[2][4] = 1$,$S[0][4] = 1$) così come anche P_0, P_1, P_4.

- Scrivere una funzione in C `void Genera(int a[DIM][DIM])` che genera una matrice quadrata booleana (valori 0 e 1) di DIM righe e colonne. La matrice deve essere simmetrica, i valori della diagonale principale devono essere tutti 1 mentre i rimanenti devono essere valori random.
- Scrivere una funzione `int Triangolo(int a[DIM][DIM], int n, int *px, int *py, int* pz)` che calcola se esiste un triangolo nell'insieme di punti e segmenti descritto dalla matrice. In caso affermativo mette i tre vertici nelle variabili px, py e pz. In caso caso non ci siano triangoli, la funzione restituisce 0.

- Scrivere un programma in C che:
  1. Usando la funzione `Genera`, genera una matrice di dimensione 15x15 e stampa tale matrice ben formattata sullo schermo.
  2. Utilizza la funzione `Triangoli` per calcolare i vertici di un triangolo, se esiste.
  3. Stampa sullo schermo le informazioni calcolate.
- Discutere la complessità degli algoritmi implementati in funzione della dimensione della matrice.
- Assumendo che nella matrice esiste una riga (o colonna) tutta di 1, l'algoritmo avrebbe una complessità diversa? E se le righe tutte di 1 fossero due?

Esercizio 2
Si scriva un programma C che richiede all'utente di inserire una sequenza di parole. La sequenza di immissione termina quando si immette una parola uguale alla prima che è stata immessa. A questo punto, viene scritto sullo schermo quante parole sono state inserite (non contando l'ultima) e qual'è la parola più lunga che è stata inserita. Non è consentito utilizzare funzioni predefinite sulle stringhe.

Esercizio 3
Si scriva una funzione `DispariPari` che prende in input una lista di interi L e restituisce una nuova lista DP che è una permutazione della lista L in cui tutti gli elementi dispari si trovano prima degli elementi pari. Inoltre gli elementi dispari e gli elementi pari si trovano in DP nello stesso ordine relativo che avevano nella lista L.
Ad esempio se $L = [0, 1, 2, 6, 7, 3, 4, 11]$ allora $DP = [1, 7, 3, 11, 0, 2, 6, 4]$.

La funzione dovrà essere di complessità lineare nella lunghezza della lista.

Si inserisca tale funzione nel codice fornito listaBase.c rinominando il file come
`CognomeNomeEse3.c`