

Laboratorio di programmazione e Informatica 1

- A.A. 2024-2025 -

Secondo-appello - 24 febbraio 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

Data una matrice quadrata di dimensione $m \times m$, l'operazione di *shift della riga i* consiste nello spostare ogni elemento della riga i nella posizione successiva, dunque "alla sua destra" (l'ultimo elemento verrà messo nella prima posizione). Qui sotto, come esempio, una matrice, la sua trasformata dopo uno shift nella riga di indice 2:

1	2	1	1
0	0	0	1
1	1	2	0
2	0	0	0

1	2	1	1
0	0	0	1
0	1	1	2
2	0	0	0

- Scrivere una funzione in C che genera una matrice random di DIM righe e DIM colonne a valori interi **pari** tra 10 e 40 e la stampa sullo schermo.
- Scrivere una funzione che data una matrice quadrata e un indice di riga, applica alla matrice l'operazione di shift su tale riga e la stampa sullo schermo.
- Scrivere un programma in C che usando le funzioni sopra descritte:
 1. Genera una matrice di dimensione 25x25 e stampa tale matrice sullo schermo.

2. Chiede all'utente di inserire un valore x tra 0 e 24 ed applica rispettivamente uno shift di riga di indice x , stampa la matrice sullo schermo e poi chiede nuovamente di inserire un altro valore. Quando l'utente inserisce "-1" il programma termina.
3. Discutere la complessità degli algoritmi implementati in funzione della dimensione della matrice.

Esercizio 2

Scrivere un programma in linguaggio C che legga una frase introdotta da tastiera. La frase contiene sia caratteri maiuscoli che caratteri minuscoli e contiene complessivamente al più 100 caratteri. Il programma deve svolgere le seguenti operazioni:

- Visualizzare la frase inserita.
- Costruire una nuova frase tale che ogni lettera vocale presente nella frase di partenza sia seguita dalla lettera **f** se la vocale è minuscola o dalla lettera **F** se la vocale è maiuscola nella nuova frase. Il programma deve memorizzare la nuova frase in una opportuna variabile.
- Visualizzare la nuova frase.

Ad esempio, se l'utente inserisce:

Oggi fInAlmentE possO sostEnerE l'esamE di LABORAtoriO

sullo schermo verrà visualizzato:

OFggif fIFnAFImefntEF pofssOF sofstEFnefrEF l'efsafmEF dif lAFBOFRAFtorifOF

Deve essere effettivamente generata una nuova stringa prima di essere stampata. Alla fine deve essere scritto di quanti caratteri la nuova stringa è più lunga di quella originale.

Esercizio 3

Una relazione definita in un insieme di n oggetti numerati da 1 a n può essere descritta da un insieme finito di coppie (a, b) con $1 \leq a, b \leq n$. 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.

1. Si implementi una funzione in C, **Relazione**, che prende in input una lista A (data come un puntatore al suo primo nodo) di coppie di interi e verifica se questa rappresenta correttamente una relazione. In particolare la funzione dovrà restituire 1 se non sono presenti coppie ripetute e 0 se sono presenti. Attenzione che la coppia (a, b) e la coppia (b, a) sono da considerare una ripetizione.
2. Variante facoltativa: (+3 punti) La funzione precedente modifica la lista eliminando gli eventuali nodi ripetuti (senza utilizzare funzioni aggiuntive). La funzione restituirà 1 se la lista non stata modificata o 0 se e stata modificata.
3. Si inserisca tale funzione nel codice fornito listaBase2app.c rinominando il file come **CognomeNomeEse3.c**