

Foglio esercizi 1

Esercizi su istruzioni condizionali e iterative.

Esercizio 1

Calcolare la media aritmetica tra N numeri x_1, \dots, x_N inseriti da tastiera, calcolata come

$$media = \frac{1}{N} \sum_{i=1}^N x_i.$$

Quando il numero inserito è uguale a 0, l'inserimento dei numeri da tastiera viene interrotto e vogliamo visualizzare a schermo la media dei numeri inseriti.

Esercizio 2

Dati due interi n e k , inseriti da tastiera, scrivere un programma per calcolare il coefficiente binomiale di n su k , calcolato come $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.

Esercizio 3

Scrivere un programma che, dato un intero inserito da tastiera, stampi i valori dei quadrati perfetti per tutti gli interi minori del numero inserito.

Esempio: assumiamo $N = 5$. Il programma deve stampare M^2 , per

$$M = 1, \dots, N \rightarrow 1^2, 2^2, 3^2, 4^2, 5^2.$$

Esercizio 4

Dati due interi n e k , calcolare la sommatoria $\sum_{i=1}^n k^i = k + k^2 + \dots + k^n$.

Esempio: assumiamo $n = 3$ e $k = 3 \rightarrow ris = k^1 + k^2 + k^3 = 3 + 9 + 27 = 39$.

Esercizio 5

Scrivere un programma che, preso un intero positivo N da tastiera, stampi la successione di Fibonacci fino al numero inserito.

L' n -esimo elemento della successione di Fibonacci è definito come:

- 0 se $i = 0$;
- 1 se $i = 1$;
- $F_{n-1} + F_{n-2}$ se $i > 1$.

Foglio esercizi 2

Esercizi sulle funzioni.

Per ogni funzione definita, testare il funzionamento all'interno del `main` su almeno un esempio, per verificare che la funzione ritorni il risultato atteso.

Esercizio 1

Scrivere una funzione che prenda in ingresso due interi a e b , con $b > 0$, e restituisca il risultato della potenza a^b .

Nota: Scrivere la funzione in modo sia ricorsivo che non.

Esercizio 2

Scrivere una funzione che prenda in ingresso tre numeri interi e restituisca il minimo tra i tre numeri.

Esercizio 3

Scrivere una funzione che prenda in ingresso le coordinate (x, y) di due punti del piano cartesiano e restituisca la loro distanza.

La distanza tra due punti A e B con coordinate rispettivamente (x_1, y_1) e (x_2, y_2) viene

calcolata come $d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Per calcolare la radice quadrata, si può utilizzare la funzione `sqrt` della libreria `math.h`. La libreria andrà importata all'inizio del programma:

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

Esercizio 4

Scrivere una funzione che riceva in ingresso due numeri di tipo `double` e restituisca 1 se la differenza tra i due numeri è minore di $\epsilon = 10^{-9}$, oppure 0 in caso contrario.

Esercizio 5

Scrivere una funzione che riceva in ingresso il prezzo iniziale di un articolo (di tipo `float`) e il valore percentuale di sconto (di tipo `int`), e restituisca il prezzo scontato.

Esercizio 6

Scrivere una funzione ricorsiva che riceva in ingresso un intero N e che ritorni il valore dell' N -esimo elemento della successione di Fibonacci.

L' n -esimo elemento della successione di Fibonacci è definito come:

- 0 se $i = 0$;
- 1 se $i = 1$;
- $F_{n-1} + F_{n-2}$ se $i > 1$.

Nota: la soluzione iterativa è richiesta nel Foglio Esercizi 1, basta includere quella soluzione nel corpo di una funzione, in modo che ritorni il valore dell' N -esimo elemento della successione, invece che stampare i primi N elementi.

Esercizio 7

Scrivere una funzione ricorsiva che riceva in input due numeri naturali a e b , ne calcoli l'MCD, seguendo la successione di ricorrenza:

- passo base: $a_0 = a, b_0 = b$
- step ricorsivo: $a_{n+1} = b_n, b_{n+1}$ è il resto della divisione $\frac{a_n}{b_n}$.

Foglio esercizi 3

Esercizi sui puntatori.

Esercizio 1

Definire le variabili intere v_1 , v_2 ed i puntatori p_1 , p_2 , p_3 .

Svolgere le seguenti istruzioni:

1. Stampare i valori di v_1 e v_2 ;
2. Stampare gli indirizzi di v_1 e v_2 ;
3. Assegnare valori diversi a v_1 e v_2 e stamparli;
4. Stampare i valori delle variabili puntate da p_1 e p_2 ;
5. Stampare gli indirizzi di p_1 e p_2 ;
6. Assegnare a p_1 l'indirizzo di v_1 ed a p_2 l'indirizzo di v_2 ;
7. Stampare gli indirizzi salvati in p_1 e p_2 ;
8. Stampare i valori delle variabili puntate da p_1 e p_2 ;
9. Incrementare v_1 e stampare il suo valore;
10. Stampare l'indirizzo contenuto da p_1 ;
11. Stampare il valore della variabile puntata da p_1 ;
12. Assegnare a p_2 l'indirizzo di v_1 ;
13. Stampare il valore della variabile puntata da v_2 ;
14. Incrementare il valore della variabile puntata da p_1 , attraverso p_1 stesso;
15. Stampare il valore della variabile puntata da p_2 ;
16. Assegnare a p_3 l'indirizzo di p_2 (p_3 pointer a pointer);
17. Stampare il valore della variabile puntata da p_3 (puntata a sua volta da p_2), tramite p_3 ;
;
18. Decrementare il valore della variabile puntata da p_3 (puntata a sua volta da p_2),
tramite p_3 .

Commentare ogni istruzione spiegando cosa sta succedendo in termini di assegnazioni, indirizzi e locazioni di memoria.

Esercizio 2

Si scriva una funzione che, dati quattro numeri interi a, b, c, d , scambi i loro valori in modo che, una volta finita l'esecuzione della funzione, si abbia: $a \leq b \leq c \leq d$.

Gli interi devono essere passati alla funzione tramite puntatore e la funzione deve ritornare void. La funzione sarà definita come segue:

```
void ordine_decrescente(int* a, int* b, int* c, int* d)
```

Nota: puoi utilizzare la funzione *scambia* dell'Esempio 5.2 della dispensa, ed implementare una funzione *minimo* che calcoli il minimo tra due interi passati tramite puntatore:

```
int min(int* a, int* b)
```

Esercizio 3

Si scriva una funzione ricorsiva, che prenda come input un intero positivo n e il puntatore della variabile H , che conterrà il seguente risultato:

$$H(n) = \sum_{i=1}^n \frac{1}{i} = 1 + \frac{1}{2} + \dots + \frac{1}{n}$$

La funzione deve essere definita nel seguente modo:

```
void somma_armonica(int n, double* h)
```

Foglio esercizi 4

Esercizi sugli array.

Esercizio 1

Considerare due array statici A e B dimensione N , ed inizializzati tramite input da tastiera. Creare tre funzioni che facciano le seguenti operazioni:

1. Per ogni indice i , sommare e moltiplicare $A[i]$ e $B[i]$ salvare i risultati in un terzo array C , all'indice i (usando lo stesso array di supporto, sovrascrivendo il valore per ogni operazione che deve essere eseguita) → la funzione prenda come input i tre array A, B, C e la loro dimensione;
2. confrontare gli elementi $A[i]$ e $B[i]$ dei due array e salvare l'elemento maggiore in $C[i]$ → la funzione prenda come input i tre array A, B, C e la loro dimensione;
3. determinare la media, il valore massimo ed il valore minimo dell'array C risultante dal punto precedente → la funzione prenda come input l'array C , la sua dimensione e tre variabili $min, max, mean$, inizializzate nel main e passate alla funzione come puntatori.

Esercizio 2

Letti in input due array A e B , rispettivamente di N ed M elementi, scrivere una funzione che salvi in un terzo array C tutti gli elementi presenti in B , ma non in A , e che ritorni la lunghezza di C .

Esercizio 3

Dati due array A e B di interi di dimensione N , letti da tastiera, scrivere una funzione che prenda in input A e B , e ne calcoli il prodotto scalare $A \cdot B$.

Il prodotto scalare è definito come: $A \cdot B = \sum_{i=1}^N A[i] * B[i]$.

Esercizio 4

Leggere un array di 10 numeri interi e scrivere una funzione che prenda come input l'array inserito, la sua dimensione e una variabile check, inizializzata nel main a 0 e passata alla funzione tramite puntatore, che sarà uguale a 0 se la sequenza inserita è palindroma, o 1 viceversa → se non cambia ad essere letta dalla prima cella all'ultima o viceversa.

Esempio: la sequenza 1234554321 è palindroma.

Esercizio 5

Dichiarare un array chiamato *primes*, e scrivere una funzione per inizializzare i suoi valori con i numeri primi (*un numero intero positivo è definito primo se ha esattamente due divisori distinti*) minori di 100 (considerare 2 come il primo numero primo → *primes*[0]=2).

Nota: per calcolare il resto della divisione tra due numeri *a* e *b* → *resto* = *a*%*b*

Esercizio 6

Dato un array *A* di 10 elementi, scrivere una funzione che modifichi l'array *A* stesso invertendo i valori.

Esempio: *A* = {1, 2, 3, 4, 5} → *A* = {5, 4, 3, 2, 1}.

Esercizio 7

Un numero è definito *triangolare* se è costituito dalla somma dei primi *n* numeri interi positivi. Di conseguenza, per un valore di *n*, possiamo trovare il numero triangolare T_n :

- $T_n = 1$, se $n = 1$;
- $T_n = n + T_{n-1}$, se $n > 1$.

Scrivere una funzione che prenda come input un array *A* e un intero *n*, e inserisca in *A*[*i*] il valore dell'*i*-esimo numero triangolare, per $i = 0, \dots, n - 1$. Scrivere la funzione in modo ricorsivo.

Esempio: 10 è il quarto numero triangolare definito dalla somma dei primi 4 numeri interi positivi $1 + 2 + 3 + 4 = 10$ ($n = 4$).