

Programmazione

Esercizi

Nota

Si consiglia di rifare in autonomia *anche* gli esercizi svolti in aula durante le lezioni.

Esercizio 1

- Scrivere un *programma* in cui:
 - si dichiarano tre variabili *reali*;
 - i valori di tali variabili vengono chiesti all'utente e letti dallo standard input;
 - si calcola il valore del minimo tra i valori delle variabili e tale valore viene assegnato a una nuova variabile.
- Scrivere il diagramma di flusso del programma precedente.

Esercizio 2

- Creare un *programma* che chieda all'utente 3 *numeri*, calcoli la deviazione standard e la stampi su standard output.
- Creare una *funzione* che calcoli la deviazione standard di 3 *numeri*, che vengono passati alla funzione come parametri. Creare il **main** nel quale viene richiamata la funzione precedente; i 3 *numeri* vengono chiesti all'utente e il risultato della funzione viene stampato a video.

Nota: La deviazione standard di n numeri $\{x_i\}_{i=1}^n$ è

$$\sigma = \sqrt{\sum_{i=1}^n (x_i - m)^2}$$

dove m è la media degli n numeri.

Esercizio 3

- Creare una *funzione* che abbia come unico parametro un *numero intero*. Se il numero è non negativo, la funzione restituisce come risultato il valore del numero incrementato di 1; se il numero è negativo, la funzione restituisce come risultato il valore del numero moltiplicato per 2.
- Creare il **main** nel quale viene richiamata la funzione precedente; il numero intero viene chiesto all'utente e il risultato della chiamata alla funzione viene stampato a video.

Esercizio 4

- Creare una *funzione* che abbia come unico parametro un *numero intero*. Se il numero è multiplo di 3, viene stampato *n multiplo di 3*; se il numero è congruo a 1 modulo 3, viene stampato *n congruo a 1 modulo 3*; se il numero è congruo a 2 modulo 3, viene stampato *n congruo a 2 modulo 3*.
Nota: Nella stampa a video, *n* è il valore del parametro, non il simbolo **n**.
- Scrivere il diagramma di flusso dell'algoritmo implementato nella funzione.

Esercizio 5

- Creare una *funzione* che calcoli il minimo di 3 *numeri interi*, che vengono passati alla funzione come parametri. Creare il **main** nel quale viene richiamata la funzione precedente; i 3 *numeri interi* vengono chiesti all'utente e il risultato della funzione viene stampato a video.
- Scrivere il diagramma di flusso dell'algoritmo implementato nella funzione.

Esercizio 6

Creare una *funzione* che prenda in input tre variabili *a*, *b*, *c* di tipo opportuno e che stampi a video le soluzioni dell'equazione $ax^2 + bx + c = 0$. Creare il **main** nel quale viene richiamata la funzione precedente; i valori delle 3 variabili che rappresentano i coefficienti dell'equazione di secondo grado vengono chiesti all'utente.

Esercizio 7

Creare una *funzione* che calcoli la somma degli *interi pari* da 1 a *n*, utilizzando lo statement **while**. Creare una *funzione* che calcoli la somma degli *interi pari* da 1 a *n*, utilizzando lo statement **for**. Creare una *funzione ricorsiva* che calcoli la somma degli *interi pari* da 1 a *n*. Il valore di *n* viene passato come parametro a ognuna delle

tre funzioni Creare il `main` nel quale vengono richiamate le funzioni precedenti; n è un parametro dato dall'utente e i risultati delle 3 funzioni vengono stampati a video.

Esercizio 8

Creare una funzione che calcoli la somma dei quadrati degli *interi* da 1 a n , utilizzando lo statement `while`. Creare una funzione che calcoli la somma dei quadrati degli *interi* da 1 a n , utilizzando lo statement `for`. Creare una funzione *ricorsiva* che calcoli la somma dei quadrati degli *interi* da 1 a n . Il valore di n viene passato come parametro a ognuna delle tre funzioni Creare il `main` nel quale vengono richiamate le funzioni precedenti; n è un parametro dato dall'utente e i risultati delle 3 funzioni vengono stampati a video.

Esercizio 9

Creare una funzione che calcoli la somma degli *interi* compresi tra a e b , utilizzando lo statement `while`. Creare una funzione che calcoli la somma degli *interi* compresi tra a e b , utilizzando lo statement `for`. Creare una funzione *ricorsiva* che calcoli la somma degli *interi* compresi tra a e b . I valori di a e b vengono passati come parametri a ognuna delle tre funzioni Creare il `main` nel quale vengono richiamate le funzioni precedenti; a e b sono chiesti all'utente e i risultati delle 3 funzioni vengono stampati a video.

Esercizio 10

Creare una funzione che calcoli a^b dove a e b sono numeri interi *interi* e $b > 0$, utilizzando lo statement `while`. Creare una funzione che calcoli a^b dove a e b sono numeri interi *interi* e $b > 0$, utilizzando lo statement `for`. Creare una funzione *ricorsiva* che calcoli a^b dove a e b sono numeri interi *interi* e $b > 0$. I valori di a e b vengono passati come parametri a ognuna delle tre funzioni Creare il `main` nel quale vengono richiamate le funzioni precedenti; a e b sono chiesti all'utente e i risultati delle 3 funzioni vengono stampati a video.

Esercizio 11

Creare la funzione `somma_geometrica_w` che calcoli, usando lo statement `while`,

$$\sum_{k=0}^n q^k \quad (1)$$

dove q e n sono parametri della funzione. Creare la funzione `somma_geometrica_f` che calcoli la somma in (1), usando lo statement `for`. Creare la funzione *ricorsiva* `somma_geometrica` che calcoli la somma in (1). Creare una funzione `stampa_geometrica`, con parametro m , che stampi il risultato di `somma_geometrica` per valori di n tra 0 e m . Creare il `main` nel quale viene richiamata la funzione `stampa_geometrica`.

Esercizio 12

Scrivere una *funzione* che prenda in input un *intero* e stampi sullo schermo **pari** se il numero è pari, **dispari** se il numero è dispari. Usare il costrutto **switch**. Creare il **main** nel quale viene richiamata la funzione precedente.

Esercizio 13

Creare la funzione **somma** che calcoli

$$4 \sum_{k=0}^n \frac{(-1)^k}{2k+1}$$

al variare del numero intero **n**. Creare la funzione **stampaSomma**, con parametro **m**, che stampi il risultato di **somma** per valori di **n** tra 0 e **m**. A quale valore si avvicina al crescere di **m**?

Esercizio 14

- Scrivere la *funzione* **somma_divisori** che prenda in input un *intero* *n* e restituisca la somma di tutti i divisori di *n* strettamente minori di *n*;
- Scrivere la *funzione* **perfetto** che prenda in input un *intero* *n* e restituisca 0 se *n* è un numero perfetto; -1 se *n* è un numero difettivo; 1 se *n* è un numero abbondante.
- Scrivere un **main** in cui viene chiesto un intero *n* all'utente. Se il numero *n* è perfetto, viene stampata la stringa **Numero Perfetto**; se il numero *n* è difettivo, viene stampata la stringa **Numero Difettivo**; se il numero *n* è abbondante, viene stampata la stringa **Numero Abbondante**. Usare il costrutto **switch**.

Nota: Un intero *n* si dice

- perfetto, se è uguale alla somma dei suoi divisori strettamente minori di *n* stesso (e.g.: $6=1+2+3$)
- difettivo, se è maggiore della somma dei suoi divisori strettamente minori di *n* stesso (e.g.: $5>1$)
- abbondante,, se è minore della somma dei suoi divisori strettamente minori di *n* stesso (e.g.: $12<1+2+3+4+6$)

Esercizio 15

Creare una funzione *ricorsiva* che, data una sequenza di numeri positivi, terminata da un numero negativo, calcoli la somma dei numeri positivi inseriti. La sequenza di numeri

è data da utente tramite tastiera. Non si conosce a priori quanti numeri saranno inseriti. Creare il **main** nel quale viene richiamata la funzione precedente e il risultato viene stampato a video.

Esercizio 16

Creare una funzione *ricorsiva* che, data una sequenza di numeri positivi, terminata da un numero negativo, calcoli la media dei numeri positivi inseriti. La sequenza di numeri è data da utente tramite tastiera. Non si conosce a priori quanti numeri saranno inseriti. Creare il **main** nel quale viene richiamata la funzione precedente e il risultato viene stampato a video.

Esercizio 17

Creare una *funzione* che prenda in input un vettore **v**[] di interi e un intero **n** e restituisca il minimo valore tra tutti gli elementi del vettore. Creare una *funzione* che prenda in input un vettore **v**[] di interi e un intero **n** e restituisca l'indice dell'elemento contenente il minimo valore tra tutti gli elementi del vettore. Creare il **main** nel quale vengono richiamate le funzioni precedenti, applicate a un vettore definito arbitrariamente all'inizio del **main**, e vengono stampati a video i risultati delle due funzioni.

Esercizio 18

Creare una *funzione* che prenda in input un vettore **v**[] di interi e un intero **n** e restituisca il numero degli elementi pari nel vettore. Creare il **main** nel quale viene richiamata la funzione precedente, applicata a un vettore definito arbitrariamente all'inizio del **main**, e viene stampato a video il risultato della funzione.

Esercizio 19

Creare una *funzione* che calcoli la deviazione standard di **n numeri** memorizzati in un array. Creare il **main** nel quale viene richiamata la funzione precedente, applicata a un vettore definito arbitrariamente all'inizio del **main**, e viene stampato a video il risultato.

Nota: La deviazione standard degli n numeri $\{x_i\}_{i=1}^n$ è

$$\sigma = \sqrt{\sum_{i=1}^n (x_i - m)^2}$$

dove m è la media degli n numeri.

Esercizio 20

Creare una *funzione* che legga una sequenza di interi positivi terminata da un numero negativo e memorizzi gli elementi (positivi) letti in un array. Gli elementi positivi letti possono essere in numero compreso tra 0 e 50 (estremi inclusi). La funzione deve restituire la media degli elementi memorizzati nell'array. Creare il **main** nel quale viene richiamata la funzione precedente.

Esercizio 21

Creare una *funzione* che consenta di leggere una sequenza di non più di 100 interi terminati da negativo, memorizzarli in un array e calcolarne il massimo. È necessario memorizzare i dati in un array per stabilire il massimo?

Esercizio 22

Creare una funzione che, dato un array di interi contenente i coefficienti di un polinomio di grado n , scriva in un nuovo array i coefficienti della derivata. Creare il **main** nel quale viene richiamata la funzione precedente e nel quale vengono stampati gli elementi di entrambi usando una funzione apposita.

Esempio: Il polinomio $3x^2 + 2x + 1$ viene rappresentato con l'array $v=\{3,2,1\}$. L'array contenente i coefficienti della derivata dev'essere $w=\{0, 6, 2\}$.

Esercizio 23

Creare una *funzione* che prenda in input due vettori della stessa lunghezza e assegni a un altro vettore la somma (elemento per elemento) dei due vettori di input. Creare una *funzione* che prenda in input due vettori della stessa lunghezza e assegni a un altro vettore il prodotto puntuale (elemento per elemento) dei due vettori di input. Creare una *funzione* che prenda in input due vettori della stessa lunghezza e restituisca il loro prodotto scalare (somma dei prodotti delle componenti omonime). Creare il **main** nel quale vengono richiamate le funzioni precedenti, applicate a due vettori definiti arbitrariamente all'inizio del **main**, e vengono stampati a video i risultati delle due funzioni.

Esercizio 24

Creare una funzione *booleana* che prenda in input due stringhe e restituisca **true** se sono uguali, **false** altrimenti. Creare il **main** nel quale vengono richieste le due stringhe all'utente, vengono lette da standard input e viene richiamata la funzione precedente. Il risultato viene stampato a video.

Nota: Non utilizzare la funzione **strcmp** definita in **cstring**.

Esercizio 25

Creare una funzione *booleana* che prenda in input due stringhe e restituisca **true** se hanno la stessa lunghezza, **false** altrimenti. Creare il **main** nel quale vengono lette le due stringhe dal file **stringhe.txt** e viene richiamata la funzione precedente. Il risultato viene stampato a video.

Nota: Non utilizzare la funzione **strlen** definita in **cstring**.

Esercizio 26

Creare una funzione che prenda in input due stringhe. All'interno della funzione viene definita una stringa di lunghezza opportuna, a cui viene assegnato il risultato della concatenazione tra le due stringhe passate come parametri alla funzione. Il risultato della concatenazione viene stampato sul file **stringhe.txt**. Creare il **main** nel quale viene richiamata la funzione precedente.

Nota: Non utilizzare la funzione **strcat** definita in **cstring**.

Esercizio 27

Creare una funzione che prenda in input due vettori e calcoli il loro prodotto scalare, sedefinito. Creare una funzione che prenda in input due vettori, supponendo che il primo sia interpretato come vettore colonna e il secondo sia un vettore riga, e calcoli il prodotto righe per colonne dei due vettori (ottenendo una matrice). Creare il **main** nel quale vengono richiamate le funzioni precedenti su due vettori definiti a piacere.

Esercizio 28

- Definire la struct **persona** con i seguenti campi: **Nome**, **Cognome**, **AnnoNascita**.
- Scrivere la funzione **analizza**, che prenda in input un array di elementi di tipo **persona** e restituisca: **true** se il campo **Cognome** di almeno un elemento dell'array corrisponde a *Verdi*; **false** altrimenti.
- Scrivere un *programma* in cui:
 - * Viene definito un array di tipo **persona** e di lunghezza 100
 - * Si inizializzano alcuni elementi dell'array con dati scelti a piacere.
 - * Viene richiamata la funzione **analizza** sull'array definito in precedenza.

Esercizio 29

- Definire la struct **studente** con i seguenti campi di tipo opportuno: **nome**, **cognome**, **matricola**, **mediaesami**.

- Scrivere la *funzione* **stampa** che prenda in input un array di elementi di tipo **studente** e che stampi i numeri di matricola di tutti gli studenti la cui media dei voti è maggiore di 27 sul file `output.txt`, su righe diverse.
- Scrivere un *programma* in cui viene dichiarato un array di nome **s**, di tipo **studente** e di lunghezza 2. Agli elementi dell'array vengono assegnati dati scelti a piacere. La funzione **stampa** viene applicata all'array **s**.

Esercizio 30

Creare una funzione che effettui l'ordinamento di un vettore, ordinando i suoi elementi dal maggiore al minore. Creare il **main** nel quale viene richiamata la funzione precedente su un vettore definito a piacere.

Nota

Altri esercizi sono stati suggeriti in aula e/o sono suggeriti nei commenti ai file C++ presenti su elly.