

RIPASSO FUNZIONI E PROCEDURE

Nei seguenti programmi devi usare le funzioni/procedure il più possibile

1. Fare un programma che letto un numero non negativo N e un carattere C, stampi a video N volte C
Per esempio se N=4 e C= 'A' a schermo compare

```
AAAA
```

2. Usando le funz/proc create nell'esercizio precedente fare un programma che letto un numero non negativo N e un carattere C, stampi a video un quadrato NxN di caratteri C (a video NON sembrerà un quadrato, ma fa lo stesso, l'importante è che ci siano N caratteri in N righe). Per esempio se N=3 e C= 'A' a schermo compare

```
AAA
```

```
AAA
```

```
AAA
```

3. Usando le funz/proc create precedentemente fare un programma che letto larghezza L e altezza H non negativi e un carattere C, stampi a video un rettangolo HxL di caratteri C. Per esempio se H=2 e L=3 e C= 'A' a schermo compare

```
AAA
```

```
AAA
```

4. Usando le funz/proc create degli esercizi precedenti stampare triangolo fare un programma che letto un numero non negativo H e un carattere C, stampare a video un triangolo rettangolo isoscele di altezza H. Per esempio se H = 3

```
  X
```

```
 XX
```

```
XXX
```

5. Usando le funz/proc create degli esercizi precedenti stampare n strisce pedonali lunghe lung se n=3 e lung = 5 saranno così

```
XXXXX
```

```
XXXXX
```

```
00000
```

```
00000
```

```
XXXXX
```

```
XXXXX
```

6. Creare la funzione *isTriangolo()* che, dati 3 numeri, restituisce true/1 se possono essere le misure di un triangolo, false/0 altrimenti. Creare un main che testi questa funzione.
7. Usando la funzione dell'esercizio precedente, scrivere la funzione *isIsoscele()* che, dati 3 numeri, restituisce true/1 se possono essere le misure di un triangolo isoscele, false/0 altrimenti. Creare un main che testi questa funzione.
8. Creare la procedura POT2(x) che eleva alla seconda il suo parametro x, cioè se inizialmente x =3 alla fine x vale 9 (utilizza il passaggio dei parametri opportuno). Creare un main che testi questa funzione.
9. Utilizzando la procedura POT2(x) e la funzione in C sqrt(x) crea la funzione IPOTENUSA() che dati i due cateti restituisce l'ipotenusa. Creare un main che testi questa funzione.
10. Crea la procedura INVERTI(x) che inverte le cifre del suo parametro che deve essere un intero positivo. Per esempio se x = 12 alla fine x=21. Creare un main che testi questa funzione.
11. Crea la procedura CRESCENTI(x,y,z) che fa in modo che in x finisca il minore dei 3 numeri, in z il maggiore e in y il numero rimanente.

RIPASSO VETTORI E STRINGHE

```
Int vett[10]
```

```
Vett = &vett
```

12. Caricare un vettore con la tabellina del numero x letto da tastiera (se x è 3 il vettore conterrà 0,3,6,9,...,30)
13. Calcolare la somma degli elementi di posizione pari e quelli di posizione dispari di un vettore (procedura con 2 parametri per indirizzo) LO ZERO è CONSIDERATO PARI
14. Dato un vettore A di interi creare i vettori PARI e DISPARI formati rispettivamente dagli elementi pari in posizione dispari e dispari in posizione pari presenti in
A [10, 12, 13, 1] PARI[12] DISPARI[13]
15. Dato un vettore di reali trovare il minimo e il massimo, quindi ordinarlo in ordine decrescente usando un algoritmo studiato e verificare che il primo e l'ultimo elemento coincidono con i valori di minimo e massimo trovati
16. Creare una stringa random 32-126 lunga n caratteri
17. Cancellare le vocali presenti in una stringa (ciao -> c), usare la funzione isVocale(car)
18. Creare upper(s), funzione che converte in maiuscolo tutti i caratteri di s. Utilizzare la funzione isMinuscolo(car)
19. Scrivere una funzione con la seguente interfaccia bool VocaliDispari(char str[]) che restituisca TRUE se il numero di vocali presenti nella stringa è dispari, FALSE altrimenti. Si utilizzi la funzione isVocale(c) per fare il confronto corretto.

MATRICI

20. Costruire la tavola pitagorica per i numeri da 0 a 10, in pratica la matrice che ha per componenti il valore che si ottiene moltiplicando l'indice della riga per l'indice della colonna.
21. I dati sugli incassi di 3 reparti di un magazzino nei primi 6 mesi sono organizzati in una matrice, il numero di riga indica il numero del reparto, il numero di colonna indica il mese. Calcolare l'incasso totale per ogni mese, l'incasso totale per ogni reparto e l'incasso totale per tutti i reparti nel periodo considerato. (programma a menù)
22. Dopo aver caricato in memoria una matrice di interi quadrata di ordine N (con N inserito da tastiera non superiore a 10), sommare gli elementi della diagonale principale e quelli della diagonale secondaria
23. Data una matrice $NR \times NC$ caricata da file, creare il vettore colonna di lunghezza NR contenente la somma delle righe della matrice
24. Data una matrice $NR \times NC$ caricata da file, creare il vettore riga di lunghezza NC contenente la somma delle colonne della matrice
25. Data una matrice rettangolare MAT ($M \times N$), di interi, già carica, calcolare quanti sono gli elementi simmetrici rispetto all'asse verticale.
26. Dire se una matrice è speculare rispetto all'asse verticale
27. Data una matrice rettangolare MAT ($M \times N$), di interi, già carica, calcolare il minimo, il massimo e la media dei valori contenuti.

STRUTTURE

28. Si scriva un programma che definisca una struttura che permetta di contenere una serie di dati Squadra (struct Squadra) del tipo:

- nome squadra (stringa di lunghezza 20) ;
- goal fatti (intero) ;
- goal subiti (intero).

Il programma deve permettere nell'ordine le seguenti operazioni:

- A. Memorizzare i dati in un vettore di strutture "campionato";
- B. Stampare a terminale tutti i nomi delle squadre che hanno fatto un numero di goal maggiore del numero dei goal subiti.
- C. Letto a terminale il nome di una squadra stampare a video i goal fatti e i goal subiti o dire che non è presente nell'elenco

Esempio di contenuto dell'array di struct:

```
{"juventus", 1, 10, 12},  
{"milan", 8, 7, 6},  
{"inter", 10, 13, 11},  
{"palermo", 2, 9, 10},  
{"lazio", 5, 8, 4},  
{"udinese", 14, 5, 7}
```

29. Un array di strutture contiene l'andamento delle quotazioni di alcune azioni in borsa nell'ultima settimana. La struttura è definita in questo modo:

- nome: stringa al massimo di 20 caratteri, senza spazi;
- valori: una sequenza di 7 float (uno per ogni giorno della settimana).

Il programma deve visualizzare sullo schermo quante azioni hanno avuto un calo superiore al 10% in una sola giornata.

Ad esempio:

- KettImport 100 98 97 96 110 109 105
- ACME 100 95 100 88 81 105 107

Il programma dovrà fornire come risultato 1, in quanto la ACME ha avuto un calo superiore al 10% fra il terzo e il quarto giorno.

30. In un concorso, i partecipanti sono sottoposti a 10 prove. I risultati del concorso sono memorizzati in un array di strutture, che contiene, per ogni concorrente, i seguenti dati:

- nome: stringa contenente al massimo 20 caratteri, compreso il terminatore;
- punteggi: una sequenza di 10 interi.
- totale: calcola la somma dei punteggi durante il caricamento e verrà usato per la classifica

Caricare i dati da tastiera e visualizzare per ciascuna prova, chi è il vincitore e con quale punteggio e quindi la classifica finale.

Esempio di contenuto dell'array di struct:

```
{"Rossi", {4,6,1,2,0,4,0,1,2,2}},  
{"Bianchi", {12,0,9,4,5,2,1,1,0,5}},  
{"Verdi", {2,3,7,9,10,4,5,9,7,2}},  
{"Astolfi", {0,2,6,1,8,5,9,10,7,8}},
```

```
{"Lorenzi", {6,9,7,1,0,0,4,5,2,1}},  
{"Franchi", {1,10,11,19,4,7,2,0,2,7}}
```

31. Un negozio di alimentari ha un archivio in cui vengono memorizzati i prodotti presenti in magazzino. Per ogni prodotto in magazzino, si dispone dei seguenti dati:

- la descrizione (stringa di al massimo 20 caratteri, incluso il terminatore);
- la quantità disponibile in magazzino (int);
- data di scadenza (tipo struttura).

Il programma deve memorizzare in un altro SCADUTI.DAT tutti i prodotti che sono da cancellare dall'archivio perché scaduti rispetto alla data letta da tastiera.

I dati vengono letti da un file NEGOZIO.DAT e la tabella deve poi essere ordinata rispetto la descrizione

Esempio di contenuto del file e dell'array di struct dopo l'ordinamento:

```
melanzane 85 2019/7/3  
olio 60 2025/5/28  
pere 150 2026/11/3  
arance 210 2020/4/12  
tonno 48 2027/10/4  
pasta 75 2021/4/12  
pomodoro 61 2015/7/31  
aceto 49 2022/4/30
```

```
{"aceto",49,2022,4,30},  
{"arance",210, 2020,4,12},  
{"melanzane",85, 2019,7,3},  
{"olio",60,2025,5,28},  
{"pasta",75, 2021,4,12},  
{"pere",150,2026,11,3},  
{"pomodoro",61,2015,7,31},  
{"tonno",48, 2027,10,4},
```

32. Gioco carte Mostri caratterizzati da un nome, forza [5..10], vita [0..100]. Gestire il gioco con un menu:

- 0-esce
- 1- crea mostro
- 2- gioca (scegli mostro)
- 3- stampa mazzo

Si devono usare anche le seguenti funzioni:

- void setVita(Mostro *m, int vita); controlla correttezza vita
- void setForza(Mostro *m, int forza); controlla correttezza forza
- int getForza(Mostro m)
- int getVita(Mostro m)
- int getLancio(Mostro m) si lancia un dado con 6 facce e si moltiplica per la forza del mostro
- Mostro creaMostro(char nome[], int vita, int forza)
- Mostro creaMostroCasuale(char nome[])

FILE

33. Dato un file di testo "numeri.dat" contenente dei numeri reali, leggerli e infine salvare nel file "risultati.dat" il valore medio. Attenzione il file può non esistere o essere vuoto
34. Aggiungere al file precedente "risultati.dat" il valore massimo tra i numeri
35. Leggere un file di testo "testo.txt" carattere per carattere, contare le parole presenti caricando ogni parola in un vettore di stringhe (supporre che non ci siano più di 100 parole di lunghezza massima 20)
36. Leggere un file di testo "testo.txt" carattere per carattere, modificarlo in modo che le parole dopo i caratteri . ! ? siano in maiuscolo e quelli dopo i caratteri , ; : () in minuscolo
37. Sia dato il file GARE.TXT così costituito (da creare e riempire):

```
mario rossi 25 (minuti impiegati a portare a termine la gara) INVIO
luca verdi 17 INVIO
loris bianchi 32 INVIO
```

Produrre in output l'atleta (cognome, nome) che ha impiegato meno tempo a portare a termine la gara.

38. Dato il file Persone.txt così composto:

```
Rossi Mario 12/03/1974 M
Verdi Marta 23/12/1977 F
Alberti Gaetano 03/04/1999 M
Zanna Bianca 07/11/1978 F
```

dopo aver definito una struttura "persona" in modo appropriato, scrivere un programma in C che permetta:

- Usando una procedura chiamata lettura(), leggere i dati delle persone dal file e metterli in un array di tipo persona chiamato elenco, visualizzando a video il risultato;
- Usando la funzione coetanei(), date due persone, dica se sono quasi coetanee (la differenza di età riferita all'anno di nascita non supera i cinque anni) o restituisca falso in caso contrario;
- Chieda all'utente i propri dati e li aggiunga al file;
- Copiare il file Persone.txt in Persone2.txt