

IIS "E. Majorana" Via Alcide De Gasperi 6, Cesano Maderno (MB)

INFORMATICA - Compiti estivi 2022

Classe 3ALS

MATRICI/VETTORI

1) Programma completo con DEV-C++

Prevedere una matrice (nome a piacere) che presenta 8 righe e 8 colonne, il programma deve eseguire:

- assegnazione casuale numero partendo da 5 al massimo di 30;
- stampa contenuto matrice;
- calcolare la somma delle somme diagonale secondaria e riga numero 5.

2) Quando si definisce una matrice quali sono gli elementi fondamentali da inserire in tale dichiarazione? Fai una descrizione sintetica di questi elementi.

3) Programma completo con DEV-C++

Nell'array `Ore_Studio` sono memorizzate le ore passate a studiare da uno studente per ogni giorno del mese (generate casualmente con valori 0-4). Calcola e visualizza

- il contenuto disordinato del vettore;
- il numero medio di ore passate a studiare nel corso del mese;
- il numero di giorni in cui non ha aperto libro, inoltre valutare il valore minimo presente all'interno del vettore;
- ordinare il vettore dal numero di ore minime a ore di studio massime.

Se risulta una media di ore pari o maggiore a 2,5 il programma deve dirti che lo studente è studioso.

Suggerimento: nella funzione `rand` inserire come MAX il valore 5.

FUNZIONI

1. Indicare le frasi corrette. (due risposte esatte)

- a. Il valore calcolato da una funzione è un parametro di input.
- b. Esistono solo funzioni senza passaggio di parametri.
- c. La chiamata di una funzione avviene mediante il suo nome.
- d. Il programma principale può richiamare più di una funzione.

2. Indicare le frasi corrette. (due risposte esatte)

Un menu:

- a. offre diverse alternative.
- b. è implementato da un ciclo for.
- c. non consente di abbandonare il programma.
- d. richiama delle funzioni.

3. Indicare le frasi corrette. (due risposte esatte)

- a. Una funzione non può essere utilizzata più volte da un programma.
- b. Possono essere utilizzate funzioni sviluppate da terze parti.
- c. Una funzione è un modulo autonomo di codice.
- d. Una funzione non può ricevere valori diversi dei parametri.

4. Indicare le frasi corrette. (due risposte esatte)

Le funzioni predefinite:

- a. accettano solo due argomenti.
- b. possono restituire un valore.
- c. non possono essere chiamate.
- d. sono funzioni di libreria.

5. Indicare le frasi corrette. (due risposte esatte)

- a. Tra parentesi è indicato il valore di ritorno.
- b. Nel fianco sinistro della funzione sono elencati i parametri.
- c. Può essere dichiarata una variabile entro la funzione.
- d. La parola chiave return specifica il valore di ritorno.

6. Indicare le frasi corrette. (due risposte esatte)

- a. Alla chiamata i parametri attuali vengono copiati in quelli formali.
- b. I parametri formali sono valori numerici.
- c. I parametri attuali sono dei meri simboli.
- d. I parametri passati a una funzione possono avere di volta in volta diverso valore.

7. Indicare le frasi corrette. (due risposte esatte)

- a. Una variabile globale è visibile a tutti i livelli.
- b. Una funzione può dichiarare al suo interno variabili globali.
- c. Una variabile locale è visibile solo nel livello nel quale è dichiarata.
- d. Una funzione non può dichiarare al suo interno più variabili locali.

8. Indicare le frasi corrette. (due risposte esatte)

- a. Nel passaggio per indirizzo viene passato un puntatore.
- b. Il passaggio per indirizzo è utilizzato nel caso di quantità esigue di dati.
- c. Nel passaggio per valore viene passato l'indirizzo iniziale del blocco di dati.
- d. Nel passaggio per indirizzo la funzione manipola direttamente i dati originari.

9. Indicare le frasi corrette. (due risposte esatte)

- a. `int a` è una dichiarazione di puntatore.
- b. `int &a` è una dichiarazione di puntatore.
- c. `puntaA = &A` risale dalla variabile al puntatore.
- d. `A = *puntaA` risale dal puntatore al valore della variabile.

Esercizi di analisi

1. Analizzare bene i seguenti programmi, concettualizzarli e quindi scrivere nella casella di testo a fianco di ogni programma il risultato della stampa a video.

a.

```
#include <iostream>
int trovamin (int A, int B )
{
    int MIN ;
    if (A < B )
        MIN = A;
    else
        MIN = B;
    return MIN;
}
int main()
{
    int RISULTATO1, RISULTATO2;
    RISULTATO1 = trovamin ( 3, 5);
    RISULTATO2 = trovamin ( 8, 7);
    printf ("%d \n", RISULTATO1);
    printf ("%d \n", RISULTATO2);
    system ("pause");
}
```



b.

```
#include <iostream>
float calcola (float NUM1, float NUM2 , char OPERAZIONE)
{
    float RESULT;
    if (OPERAZIONE == '+') RESULT = NUM1 + NUM2;
    if (OPERAZIONE == '-') RESULT = NUM1 - NUM2;
    if (OPERAZIONE == '*') RESULT = NUM1 * NUM2;
    if (OPERAZIONE == '/') RESULT = NUM1 / NUM2;
    return RESULT;
}

int main()
{
    float RISULTATO1, RISULTATO2;
    RISULTATO1 = calcola (3, 5, '*');
    RISULTATO2 = calcola (8, 7, '-');
    printf ("%5.1f \n", RISULTATO1);
    printf ("%5.1f \n", RISULTATO2);
    system ("pause");
}
```



```

c. #include <iostream>
char contavoti (int *ca, int *ma)
{
    int i, votiCarlo = 0, votiMario = 0;
    for (i = 0; i < 5; i = i + 1)
    {
        votiCarlo = votiCarlo + ca[i];
        votiMario = votiMario + ma[i];
    }

    if (votiCarlo > votiMario) return 'C';
    if (votiMario > votiCarlo) return 'M';
    if (votiCarlo == votiMario) return 'p';
}

int main ()
{
    int Carlo[5] = {7, 6, 8, 5, 9}, Mario[5] = {6, 6, 7, 5, 7};
    char migliore;
    migliore = contavoti (Carlo, Mario);
    if (migliore == 'C') printf ("Carlo \n");
    if (migliore == 'M') printf ("Mario \n");
    if (migliore == 'p') printf ("pari \n");
    system ("pause");
}

```



STRUTTURE

1. Indicare le frasi corrette. (due risposte esatte)

Sia data la seguente struttura:

```

struct DISCO {
    char CANTANTE[30]; //nome del cantante
    char CANZONE[30]; //nome della canzone
    int COPIE; //copie vendute della canzone
}

```

- a. CANTANTE è un campo.
- b. CANTANTE è una stringa.
- c. DISCO è una variabile.
- d. CANZONE non è un campo.

2. Indicare le frasi corrette. (due risposte esatte)

Data la seguente dichiarazione: struct DISCO DISCO1;

- a. printf("%d", CANZONE) visualizza la canzone contenuta in DISCO1.
- b. printf("%c", DISCO1.CANTANTE[0]) visualizza l'iniziale del cantante.
- c. printf("%d", COPIE) visualizza le copie vendute da DISCO1.
- d. è possibile dichiarare anche la variabile struct DISCO DISCO2.

3. Indicare le frasi corrette. (due risposte esatte)

Data la seguente dichiarazione: `struct DISCO DISCHI[100];`

- a. `struct DISCO` è una variabile.
- b. `DISCHI` è una variabile.
- c. il vettore di struttura ha estensione 100.
- d. ogni casella di `DISCHI` contiene un campo.

4. Indicare le frasi corrette. (due risposte esatte)

Data la seguente dichiarazione: `struct DISCO DISCHI[100];`

- a. `DISCO[0].CANTANTE` è il secondo cantante.
- b. `DISCHI[1].CANZONE` è la seconda canzone.
- c. `DISCHI[2].COPIE` individua un campo numerico.
- d. `DISCHI[100].CANZONE` è la 100-esima canzone.

5. Indicare le frasi corrette. (due risposte esatte)

Data la seguente dichiarazione: `struct DISCO DISCHI[100];`

- a. `for (i = 0; i < 100; i = i + 1) printf ("%c", DISCHI[i].CANTANTE[0])` visualizza tutte le iniziali dei cantanti.
- b. `for (i = 0; i < 100; i = i + 1) printf ("%c", DISCO[i].CANZONE[0])` visualizza tutte le iniziali delle canzoni.
- c. `for (i = 0; i < 100; i = i + 1) gets (DISCHI[i].CANTANTE)` acquisisce i nomi dei cantanti.
- d. `for (i = 0; i < 100; i = i + 1) printf ("%c", DISCHI[i].COPIE)` visualizza tutte le copie vendute.

Esercizi di analisi

1. Analizzare bene i seguenti programmi, concettualizzarli e quindi scrivere a fianco di ogni programma il risultato della stampa a video.

Gli esercizi fanno riferimento alla seguente struttura; il significato dei suoi campi è specificato nei commenti a lato.

```
struct COMPUTER {  
    char MARCA[30];           /* marca del computer */  
    char MODELLO[30];        /* nome del modello */  
    int ANNOPRODUZIONE;      /* anno di produzione */  
    long int QUANTITA;        /* quantità di computer prodotti di dato Modello e Marca */  
    char PORTAUSB;           /* campo carattere che specifica se è presente una porta USB:  
                             vale 's' o 'n' se è presente o no */  
}
```

```

a. struct COMPUTER PC;
   strcpy(PC.MARCA, "Acer");
   strcpy(PC.MODELLLO, "Aspire");
   PC.ANNOPRODUZIONE = 2014;
   PC.QUANTITA = 100500;
   PC.PORTAUSB = 's';
   printf ("La marca del PC e' : %s \n", PC.MARCA);
   printf ("Il modello del PC e' : %s \n", PC.MODELLLO);
   printf ("L'anno di produzione e' : %d \n", PC.ANNOPRODUZIONE);

b. struct COMPUTER PC[5] = {"Acer", "Aspire", 2014, 100500, 's', "Packard",
   "Easynote", 2013, 100000, 's', "Toshiba", "Satellite", 2010, 150000, 'n', "Sony",
   "Vaio", 2012, 57000, 'n', "IBM", "Thinkpad", 2014, 130000, 'n'};
   int i;

   for (i = 0; i < 5; i = i + 1)
       if ( PC[i].QUANTITA > 100000) {
           printf ("%s", PC[i].MARCA);
           printf ("\n");
       }

c. struct COMPUTER PC[5]= {"Acer", "Aspire", 2014, 100500, 's', "Packard", "Easynote",
   2013, 100000, 's', "Toshiba", "Satellite", 2010, 150000, 'n', "Sony", "Vaio", 2012,
   57000, 'n', "IBM", "Thinkpad", 2014, 130000, 'n'};
   int i, Confronto;

   for (i = 0; i < 5; i = i + 1) {
       Confronto = strcmp (PC[i].MODELLLO, "Satellite");
       if ( Confronto==0) {
           printf ("La marca del PC e' : %s \n", PC[i].MARCA);
           printf ("Il modello del PC e' : %s \n", PC[i].MODELLLO);
           printf ("L'anno di produzione e' : %d \n", PC[i].ANNOPRODUZIONE);
       }
   }

```