Carrazza →
Mereghetti →
Spoletini →
Tamascelli →

INFORMATICA – 8 febbraio 2019

Cognome _____ Nome _____ Matr _____

Lab (voto/quando) _____/_____ Firma _____

1) Sia l'array **int A[6] = {1, 2, 3, 4, 5, 6}**. Scrivete in maniera alternativa gli assegnamenti indicati

```
*A = 5;                                  _____
*(A + *(A)) = A[5];                      _____
A[A[0]] = A[5];                          _____
A[3] = A[5] + A[4];                      _____
```

2) Sia il frammento di codice

```
int k = 0;
for ( int i = 0; i < 3; i++ )
      for ( int j = 0; j <= i; j++ )
            k = k + i*j;
```

Quante volte viene eseguito l'assegnamento **k = k + i*j** ? _____

Quale sarà il valore finale di **k** ? _____

3) Eseguite il codice sottostante e stabilite il valore delle variabili **a, b, c**.

```
int matr = <il vostro numero di matricola di 6 cifre>;
float a = ((matr/10)*10.0)/2;            a = _____
float b = ((matr/10)*10)/2;              b = _____
int c = ((matr/10)*10.0)/2;              c = _____
```

4) La struttura

```
struct esame {
      string nome;
      int cfu, voto;
};
```

rappresenta un esame superato da uno studente, memorizzando nei suoi tre campi, rispettivamente, il nome dell'esame (es. **Informatica**), il relativo numero di cfu (es. **6**) ed il voto conseguito (es. **27**). Scrivete un frammento di codice che chieda all'utente quanti esami superati vuole inserire e dimensioni di conseguenza un *array dinamico* di tipo **esame**. Successivamente il codice deve leggere l'uno dopo l'altro gli esami superati (assumete che i dati inseriti siano sempre corretti). Al termine, il codice deve stampare la *media pesata* dei voti conseguiti e la descrizione degli esami in cui è stato conseguito il *voto maggiore* (che, attenzione!!, non è necessariamente 30).

Ricordiamo che, data una sequenza di voti **v1, v2, ..., vn** conseguiti ad esami di cfu **c1, c2, ..., cn**, la *media pesata* di tale sequenza di voti si calcola come **(c1*v1 + c2*v2 + ... + cn*vn)/(c1 + c2 + ... + cn).**

(SVOLGIMENTO ALLA PAGINA SEGUENTE)

(SVOLGIMENTO ESERCIZIO 4)

5) Scrivete la funzione

bool notOccur(int *X, int *Y, int dim)

che accetta in ingresso due array di interi **X** e **Y** di dimensione **dim** e restituisce **true** se ogni elemento di **X** *non* compare in **Y**, **false** altrimenti.

Esempio: assumendo **X = {5, 6, 28, 4, 13, 13}** e **Y = {1, 5, 7, 8, 13, 4}** l'invocazione **notOccur(X, Y, 6)** restituisce **false** poiché **X** e **Y** hanno elementi in comune.