

Programmazione I / Informatica generale

Prova scritta - 2 Luglio 2008

NOTA: Nei programmi si trascuri ogni problema legato al tipo ed al valore di ritorno della funzione **main** e si sottintenda la presenza delle direttive

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std ;
```

Infine, laddove si afferma che un programma o frammento di codice fornisce un certo risultato, è da intendersi che, in accordo alle regole del linguaggio, tale programma o frammento di codice fornisce quel risultato per qualsiasi esecuzione su qualsiasi macchina.

PARTE 1 – RISPOSTA SINGOLA - Ogni domanda ha una sola risposta VERA.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
- Una risposta lasciata in bianco viene calcolata: 0

1. (3, -0.5) Dato il seguente programma:

```
main()
{
    int i = 10 ;
    char s[10] ;
    cin>>i ; cin>>s ;
    cout<<i<<" "<<s<<endl ;
}
```

- a) se l'utente inserisce su *stdin* **2 prova** (c'è uno spazio tra il carattere 2 e la stringa *prova*) e poi preme INVIO, l'istruzione *cin>>i* manda lo stream *cin* in stato di errore perché legge una sequenza di caratteri che non rappresenta nessun numero
- b) se l'utente inserisce su *stdin* **prova 2** (c'è uno spazio tra la stringa *prova* ed il carattere 2) e poi preme INVIO, il programma stampa
2 prova
- c) per memorizzare correttamente il valore 2 e la stringa *prova* nelle variabili *i* ed *s*, l'unica soluzione è che l'utente inserisca su *stdin* il carattere **2**, poi preme INVIO, quindi scriva **prova** e preme di nuovo INVIO
- d) nessuna delle altre risposte è corretta

2. (3, -0.5) Il seguente codice:

```
int c;
while ((c = cin.get()) != EOF)
    cout.put(c);
```

- a) Esegue il ciclo **while** all'infinito
- b) Copia ciò che viene inserito da standard input sullo standard output
- c) Non effettua alcuna operazione poiché non è specificato un file *stream* da cui leggere i dati
- d) Non riproduce gli spazi sullo standard output

3. (3, -0.5) Dato il seguente programma e sapendo che se un operatore binario ha un operando di tipo **unsigned int** e l'altro di tipo **int**, entrambi sono convertiti ad **unsigned int** prima di eseguire la relativa operazione:

```
main() {  
    unsigned int i = 5 ;  
    for (int j = 0 ; i - j >= 0 ; j++)  
        cout<<i-j<<endl ;}
```

- a) Esegue un ciclo infinito
 - b) Stampa i numeri da 5 a 0
 - c) Stampa i numeri da 5 a 1
 - d) Nessuna delle altre risposte è corretta
4. (3, -0.5) Dato il seguente programma:
- ```
void fun(int) ;
main()
{
 char car = 'a' ;
 fun(car) ;
}

void fun(int a)
{
 cout<<a ;
}
```
- a) Se eseguito, il programma stampa il carattere **a** su *stdout*
  - b) Il programma contiene un errore di gestione della memoria
  - c) Nessuna delle altre risposte è corretta
  - d) Il programma termina prima di stampare qualcosa su *stdout*
5. (2, -0.5) La seguente istruzione  
**int b = static\_cast<double>(5)/2+3.0/2;**  
fornisce:
- a) un errore a tempo di compilazione perché il risultato dell'espressione non è rappresentabile mediante il tipo **int**
  - b) il valore 4 nella variabile **b**
  - c) un fallimento a tempo di esecuzione perché il risultato dell'espressione non è rappresentabile mediante il tipo **int**
  - d) il valore 3 nella variabile **b**

**PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE -  
Ogni domanda può avere una o più risposte CORRETTE.**

- Ogni risposta esatta viene calcolata: +1
  - Ogni risposta errata viene calcolata: -0.5
  - Una risposta lasciata in bianco viene calcolata: 0
6. Un oggetto allocato dinamicamente all'interno di una funzione
- a) occupa memoria dall'inizio alla fine del programma;
  - b) è effettivamente allocato solo quando l'istruzione di allocazione viene eseguita;
  - c) smette di occupare spazio in memoria quando viene deallocato;
  - d) se non esplicitamente deallocato prima, è implicitamente deallocato al termine della funzione.

7. Dato il seguente programma:

```
main()
{
 enum stato {pieno, vuoto} ;
 stato v[3] = {pieno, pieno, vuoto} ; int i ;
 cin>>i ;
 if ((i >= 0 && i < 3) && v[i] != pieno) {
 v[i] = pieno ;
 cout<<"Riempito" ;
 }
}
```

- a) Se l'utente immette 2 su *stdin* il programma stampa **Riempito**
  - b) Il programma contiene un errore logico
  - c) Il programma contiene un errore di accesso alla memoria
  - d) Se l'utente immette 3 su *stdin* il programma non stampa nulla e termina correttamente
8. Il seguente programma:

```
struct parola {char *stringa ; int lun ;} ;

main()
{
 parola v[2] ;
 for (int i = 0 ; i < 2 ; i++) {
 cin>>v[i].stringa ;
 v[i].lun = strlen(v[i].stringa) ;
 cout<<v[i].stringa<<" "<<v[i].lun<<endl ;
 }
}
```

- a) legge correttamente da *stdin* due parole e stampa su *stdout* ciascuna delle due parole seguita dalla rispettiva lunghezza
- b) contiene un errore di gestione della memoria
- c) può comportarsi in modo non prevedibile
- d) fallisce sempre prima di aver stampato le due parole

9. Nel seguente programma:

```
1:int a = 1;
2:
3:fun()
4:{
5: int i ;
6: for (i = 0 ; i < 5 ; i++) {
7: float a = i / 2.0 ;
8: cout<<a * i ;
9: }
10: cout<<a + i ;
11:}
12:
13:main()
14:{
15: float b = 1.4 ;
16: fun() ;
17: cout<<a + b<<endl ;
18:}
```

- a) la variabile **a** definita alla riga 1 ha tempo di vita pari all'intero programma;
- b) la variabile **a** definita alla riga 1 non ha visibilità pari all'intero programma;
- c) la variabile **a** definita alla riga 1 non è più visibile alla riga 17 dopo l'esecuzione della funzione **fun**;
- d) la variabile **a** definita alla riga 7 non è visibile fino alla fine della funzione **fun**.

### PARTE 3 – DOMANDE APERTE –

- **Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda**
  - **Una risposta errata causa una penalità che dipende dalla gravità dell'errore, ed al più uguale al punteggio negativo riportato a fianco della domanda**
  - **Una risposta lasciata in bianco viene calcolata: 0**
10. (5 pt.) Scrivere una funzione che prenda in ingresso sia un vettore di  $N$  interi, implementato mediante un array di interi di lunghezza maggiore o uguale di  $N$ , che un valore  $v$ . Se il vettore contiene almeno un elemento di valore  $v$ , la funzione elimina al più un elemento di tale valore dal vettore stesso, altrimenti la funzione elimina dal vettore l'elemento di valore massimo.

11.(5 pt) Cosa viene scritto su *stdout* dal seguente programma C++?

```
int fun(int a[], int i, int j)
{
 int h ;
 for (h = i ; h < j ; h++)
 if (a[h] > a[h+1])
 break ;
 if (h == i)
 return 0 ;
 for (; h < j ; h++)
 if (a[h] < a[h+1])
 break ;
 return h ;
}

main()
{
 const int N = 4 ;
 int v[N] = {21, 3, 123, 45} ;
 for (int i = 0 ; i < N ; i++)
 for (int j = N - 1 ; j > i ; j--)
 if(fun(v, i, j) > 0) {
 for (int k = i ; k <= j ; k++)
 cout<<v[k]<<" " ;
 cout<<endl ;
 }
}
```



# Programmazione I / Informatica generale

## Prova scritta - 2 Luglio 2008

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_ Corso di Laurea: \_\_\_\_\_

**Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 10 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Avete due copie di questa pagina, usatene una per calcolare il voto da sole/soli durante la correzione e decidere se consegnare.**

|   | Risposte |   |   |   | Punti/<br>Penalità |      |
|---|----------|---|---|---|--------------------|------|
|   | A        | B | C | D |                    |      |
| 1 |          |   |   |   | 3                  | -0.5 |
| 2 |          |   |   |   | 3                  | -0.5 |
| 3 |          |   |   |   | 3                  | -0.5 |
| 4 |          |   |   |   | 3                  | -0.5 |
| 5 |          |   |   |   | 2                  | -0.5 |
| 6 |          |   |   |   |                    |      |
| 7 |          |   |   |   |                    |      |
| 8 |          |   |   |   |                    |      |
| 9 |          |   |   |   |                    |      |

**Risposta alla domanda 10 (5 pt):**

**Risposta alla domanda 11 (5 pt):**



# Programmazione I / Informatica generale

## Prova scritta - 2 Luglio 2008

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_ Corso di Laurea: \_\_\_\_\_

**Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 10 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Avete due copie di questa pagina, usatene una per calcolare il voto da sole/soli durante la correzione e decidere se consegnare.**

|   | Risposte |   |   |   | Punti/<br>Penalità |      |
|---|----------|---|---|---|--------------------|------|
|   | A        | B | C | D |                    |      |
| 1 |          |   |   |   | 3                  | -0.5 |
| 2 |          |   |   |   | 3                  | -0.5 |
| 3 |          |   |   |   | 3                  | -0.5 |
| 4 |          |   |   |   | 3                  | -0.5 |
| 5 |          |   |   |   | 2                  | -0.5 |
| 6 |          |   |   |   |                    |      |
| 7 |          |   |   |   |                    |      |
| 8 |          |   |   |   |                    |      |
| 9 |          |   |   |   |                    |      |

**Risposta alla domanda 10 (5 pt):**

**Risposta alla domanda 11 (5 pt):**