

**Programmazione I / Informatica Generale - Autovalutazione**  
**Prova scritta**  
**19 Novembre 2008**  
**Tempo a disposizione: 1h30min**

**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 trovi l'affermazione 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, -1) Il seguente programma:

```
int &fun(int a)
{
    int b = 2 * a ;
    return b ;
}
main()
{
    int &p1 = fun(2) ;
    int &p2 = fun(3) ;
    cout<<p1 ;
}
```

- a) contiene un errore di gestione della memoria;
- b) stampa 2;
- c) stampa 4;
- d) nessuna delle precedenti risposte è vera.

2. (+3, -0.5) Il seguente programma:

```
double a = 3 ;

int fun1(double &b, int n)
{
    a += 1 ;
    b -= n ;
    return n ;
}

void fun2(double &a)
{
    fun1(a, 5) ;
}

main()
{
    double b = 3.0 ;
    fun1(b, 2) ;
    fun2(b) ;
    cout<<a<<" "<<b<<endl ;
}
```

- a) contiene un errore di gestione della memoria;
- b) stampa 5 -4
- c) stampa 5 1
- d) nessuna delle precedenti risposte è vera

3. (+1, -1) Il seguente programma:

```
void dividi(double b, int n)
{
    b /= n ;
}

main()
{
    int b = 3 ; int n = 2 ;
    dividi(b, n) ;
    cout<<b<<endl ;
}
```

- a) stampa 1
- b) stampa 1.5
- c) contiene un errore di gestione della memoria
- d) nessuna delle precedenti risposte è corretta

4. (+2, -1) Il seguente frammento di codice:

```
int i;
cin>>i;
if (static_cast<char>(i) > 'w')
    cout<<"Maggiore" ;
```

- a) stampa o meno **Maggiore** a seconda del valore assunto dalle variabili **i** ed **w**
- b) stampa **Maggiore** per ogni valore di **i** maggiore del valore della costante carattere **'w'**
- c) contiene un errore di gestione della memoria
- d) nessuna delle precedenti risposte è corretta

5. (+2, -0.5) Data la seguente definizione:  
`char c = static_cast<int>('a') ;`
- si ha perdita di informazione
  - si inizializza c con il valore della costante carattere 'a'
  - se si tentasse di stampare il valore di c mediante `cout<<c`; si stamperebbe il codice ASCII del carattere 'a' in base 10
  - nessuna delle precedenti risposte è corretta
6. (+2, -0.5) Il seguente programma
- ```
int calcola_somma(int a[], int n)
{
    int somma = 0 ;

    for (int i = 0 ; i < n ; i++)
        somma += a[i] ;

    return somma ;
}

main()
{
    int b[] = {1, 4, 3} ;

    calcola_somma() ;
    cout<<somma<<endl ;
}
```
- stampa 0
  - stampa 8
  - contiene un errore di gestione della memoria
  - nessuna delle precedenti risposte è corretta
7. (+2; -0.5) Il seguente frammento di codice
- ```
int i=5;
for( ;i>0; ) {
    if (i==2)
        break;
    cout<<i<<" ";
    i--;
}
```
- stampa gli interi tra 5 e 3;
  - stampa gli interi tra 5 e 1 tranne il 2;
  - stampa gli interi da 5 in su;
  - non stampa alcunché poiché la `cout<<i<<" "`; interna al ciclo non viene mai eseguita.

**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

8. Il seguente programma

```
void fun(int a[], int n) // prestare attenzione ai valori degli indici
{
    for (int i = 0 ; i < n ; i++)
        a[i] = a[i+1] * 2 ;
}
```

```
main()
{
    const int M = 5 ;
    int a[] = {3, 7, 11, 2, 3} ;
    fun(a, M) ;
    for (int i = 0 ; i < M ; i++)
        cout<<a[i]<<" " ;
}
```

- a) stampa 14 22 4 6 3
- b) stampa 14 22 4 6 6
- c) contiene un errore logico
- d) contiene un errore di gestione della memoria

9. Dato il seguente programma

```
int x=0;
```

```
void funct1()
{
    x = x + 3;
}
```

```
void funct2(int x)
{
    x-- ;
    funct1();
    cout<<x<<" ";
}
```

```
main()
{
    x++;
    cout<<x<<" ";
    funct2(x);
    cout<<x;
}
```

- a) viene stampato 1 0 4
- b) viene stampato 1 0 3
- c) la funzione **funct1** incrementa di 3 il valore della variabile **x** stampata nell'ultima istruzione della funzione **main**
- d) nessuna delle precedenti risposte è corretta

10. Data la definizione

```
char a = '3';
```

ed assumendo che il tipo char sia senza segno,

- a) l'istruzione `cout<<static_cast<char>(a - 2) ;` stampa 1
- b) `static_cast<char>(a - 3)` è il codice di un carattere ancora stampabile
- c) l'istruzione `cout<<static_cast<char>(a - '1')` ; stampa 2
- d) è vero che `a >= 3`

11. Dato il seguente programma:

```
void fun(int) ;

main()
{
    int g = 3;
    fun(g) ; cout<<g ;
}

void fun(int a)
{
    int g ;
    cout<<(g++)<<" " ;
}
```

- a) Se eseguito, il programma stampa 3 4
- b) Se eseguito, il programma stampa 3 3
- c) All'interno della funzione **fun** si stampa e si modifica il valore della variabile **g** definita nella funzione **fun**
- d) Il programma causa un errore a tempo di compilazione

12. Dato, il seguente programma

```
int g = 10 ;

bool fun(int a)
{
    if (a > 10) {
        ++g ;
        return true ;
    }
    return false ;
}

main()
{
    const int m = 22 ;

    if ((m % 2 == 0) || fun(m))
        cout<<-g ;
    else
        cout<<g ;
}
```

- a) Se eseguito, il programma stampa -10
- b) Se eseguito, il programma stampa -11
- c) A parte l'inizializzazione, il valore della variabile globale **g** non è mai modificato
- d) Il valore della variabile globale **g** è modificato in conseguenza della invocazione della funzione **fun** all'interno del **main**

13. Dato il seguente programma e facendo attenzione alle effettive operazioni eseguite in base al tipo degli operandi

```
int g = 1 ;
```

```
int fun()
{
    ++g ;
    return 5 ;
}
```

```
main()
```

```
{
    const int m = 1 ;

    double h = 2 + (m * static_cast<double>(fun()) / 2) ;
    cout<<static_cast<int>(h * g) ;
}
```

- a) Se eseguito, il programma stampa 9
- b) Se eseguito, il programma stampa 4
- c) Nell'ultima istruzione del **main**, la conversione esplicita non causa alcuna perdita di informazione
- d) Il programma contiene un errore di gestione della memoria

### 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

14. [3 pt] Cosa viene stampato a video dal seguente programma C++?

```
int funz(int a, int b);
int r = 0;
```

```
main()
```

```
{
    int n1= 10, n2= 20, ris=0;
    cout<<n1<<" , "<<n2<<endl;
    ris = funz(n1, n2) + r ;
    cout<<n1<<" , "<<n2<<" , "<<ris<<endl;
}
```

```
int funz(int a, int b)
```

```
{
    r = (a + b)%2;
    if (r == 0)
        r = (a + b)/2;
    cout<<r<<endl ;
    return r+1;
}
```

**15. [3 pt] Scrivere una funzione che prenda in ingresso un array di interi ed inverta l'ordine degli elementi di tale array.**

**Programmazione I / Informatica Generale – Autovalutazione**  
**Prova Scritta**  
**19 Novembre 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.**

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					+3	-1
2					+3	-.5
3					+1	-1
4					+2	-1
5					+2	-.5
6					+2	-.5
7					+2	-.5
8						
9						
10						
11						
12						
13						

**Risposta alla domanda 14:**

**Risposta alla domanda 15:**

