

# Programmazione I / Informatica generale

## Prova scritta - 10 febbraio 2009 - 1h30min

NOTA: Nei programmi si trascuri ogni problema legato al tipo ed al valore di ritorno della funzione **main**, inoltre si sottintenda la presenza delle direttive `#include <iostream> / #include <fstream> / using namespace std ;` e non si prenda come un buon esempio la formattazione utilizzata (spesso compressa per motivi di spazio). Si interpreti “terminazione forzata”, come l'abbreviazione di “terminazione forzata del programma da parte del sistema operativo”. Infine, laddove si trovi l'affermazione che un programma o frammento di codice produce un certo risultato, è da intendersi che, in accordo alle regole del linguaggio, tale programma o frammento di codice produce 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() {
    struct s {int a, b, c ;} s1[3] ;
    ifstream f("dati.dat"); // supponiamo abbia successo
    f.read(reinterpret_cast<char *>(s1), sizeof(s) * 4) ;
    for(int i = 0 ; i <= 3 ; i++)
        cout<<s1[i].a<<" "<<s1[i].b<<" "<<s1[i].c<<endl ;
    cout<<"Finito"<<endl ;
}
```

ed assumendo che nel file **dati.dat** siano memorizzati l'uno dopo l'altro, ed in formato binario, 4 oggetti di tipo **s**

- a) il programma stampa il contenuto dei primi 3 oggetti di tipo **s** memorizzati nel file
- b) il programma stampa il contenuto dei 4 oggetti di tipo **s** memorizzati nel file
- c) il programma contiene un errore di gestione della memoria
- d) nessuna delle altre risposte è corretta

2. (3, -0.5) Dato il seguente programma:

```
int g = 1 ;

double fun() { ++g ; return g + 3 ; }

main()
{
    const int m = 2 ;

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

- a) Se eseguito, il programma stampa 9
- b) Se eseguito, il programma stampa 8
- c) Il programma stampa un numero diverso da 8 e da 9
- d) Nessuna delle altre risposte è vera

3. (2, -0.5) Dato il seguente frammento di codice

```
int a[3] ; delete [] a ; a[0] = 3 ;
```

- a) il frammento non contiene errori di gestione della memoria;
- b) l'assegnamento `a[0] = 3;` comporta un errore di accesso alla memoria perché la memoria dinamica occupata da `a` è stata deallocata;
- c) l'oggetto `a` non appartiene alla classe di memorizzazione a cui può essere applicato l'operatore `delete`;
- d) nessuna delle altre affermazioni è vera.

4. (2, -1) Data la seguente definizione:

```
int i = static_cast<char>(static_cast<int>('b') + 1) ;
```

- a) l'inizializzazione presente nella definizione può causare problemi di *overflow*
- b) una successiva istruzione `cout<<i;` stamperebbe `c`
- c) una successiva istruzione `cout<<i;` stamperebbe il codice del carattere `c`
- d) nessuna delle altre risposte è corretta

5. (2, -1) Il seguente programma:

```
void fun(int c) { c = 1 ; }
main() { int a = 3 ; int &b = a ; fun(b) ; cout<<a ; }
```

- a) contiene un errore di gestione della memoria
- b) stampa 3
- c) stampa 1
- d) nessuna delle altre affermazioni è vera

## 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. Dato il seguente programma:

```
const int N = 2000 ;
struct ss { int b; char a[2000] ; } ;
void fun(ss & d) { cout<<d.a[N/2]<<" "; d.b = 3 ; }
main()
{
    ss c ; c.b = 12 ; c.a[N/2] = '2' ; cout<<c.b<<" " ;
    fun(c) ; cout<<c.b ;
}
```

- a) Il passaggio di `c` per riferimento non permette a `fun` di modificare `c`
- b) Il passaggio di `c` per riferimento ha come costo la copia nel parametro formale `d` del contenuto di `c` stesso
- c) Il programma stampa 12 2 3
- d) L'esecuzione di `fun(c)` ; ha effetti collaterali sulla porzione della funzione `main` che segue l'invocazione della funzione `fun` stessa

7. Indicare quale/i delle seguenti espressioni booleane è/sono vera/e se `int a=2, b=0, c=3;`

- a) `c||(a+b)`
- b) `b || (c && b == 0 && a < b+c)`
- c) `a < b`
- d) `!(c && (a < b))`

8. Dato il seguente programma:

```

1:int a = 2;
2:
3:main()
4:{
5:    for (int i = 3 ; i > 1 ; i--) {
6:        float a = i / 2.0 ;
7:        cout<<static_cast<int>(a * i)<<" ";
8:    }
9:    cout<<static_cast<int>(a) ;
10:}

```

- il programma stampa **4 2 1**
- la variabile **a** definita alla riga 1 non è visibile solo in parte della funzione **main**
- il programma stampa **4 2 2**
- la definizione alla riga 6 nasconde, fino alla riga 9 inclusa, la variabile **a** definita alla riga 1

9. Dato un programma scritto in linguaggio C/C++

- Quando inizia l'esecuzione del programma vengono allocati in memoria i record di attivazione di tutte le funzioni in esso definite
- Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili locali della funzione (se presenti)
- Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) i parametri formali della funzione (se presenti)
- Quando una funzione ne invoca un'altra, viene prima deallocato il record di attivazione della funzione chiamante e quindi creato quello della funzione chiamata

### PARTE 3 – DOMANDE APERTE –

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore
- Una risposta lasciata in bianco viene calcolata: 0

10. (5 pt) Cosa viene stampato su *stdout* dal seguente programma?

```

void fun(char b[], int n)
{
    for (int i = 0 ; b[i+n] != '\0' ; i++)
        if (b[i] == '\0' || b[i] < b[i+n])
            b[i] = b[i+n] ;
}

main()
{
    char a[] = {'a', 'f', '\0', 'c', 'm', 'h', '\0'} ;
    cout<<a<<endl ;
    fun(a, 0) ;
    cout<<a<<endl ;
    fun(a, 5) ;
    cout<<a<<endl ;
    fun(a, 3) ;
    cout<<a<<endl ;
}

```

11. **(5 – 7 pt.)** Scrivere una funzione che prenda in ingresso un vettore di  $N$  interi ed elimini dal vettore tutti gli elementi di valore uguale a quello di un ulteriore parametro di ingresso  $v$ . Se ritenuto opportuno scrivere anche eventuali funzioni di supporto a quella principale. Il punteggio ottenuto varia tra 5 e 7 in funzione dell'efficienza della soluzione realizzata.

# Programmazione I / Informatica generale

## Prova scritta - 10 febbraio 2009

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/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					2	-0.5
4					2	-1
5					2	-1
6						
7						
8						
9						

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

**Risposta alla domanda 11 (5 – 7 pt):**



# Programmazione I / Informatica generale

## Prova scritta - 10 febbraio 2009

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/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					2	-0.5
4					2	-1
5					2	-1
6						
7						
8						
9						

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

**Risposta alla domanda 11 (5 – 7 pt):**