

# Programmazione I / Informatica generale

## Prova scritta

### 29 Gennaio 2008

NOTA: Si trascuri ogni problema legato al tipo ed al valore di ritorno della funzione **main**. Inoltre, nei programmi è da sottintendere la presenza delle direttive

```
#include <iostream>
#include <fstream>
using namespace std ;
```

#### 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. (2, -1) Dato il seguente programma

```
main() {
    struct s {int a, b, c ;} s1[2] ;
    ifstream f("dati");
    f.read(reinterpret_cast<char *>(s1), sizeof(s)) ;
    for(int i = 0 ; i < 2 ; i++)
        cout<<s1[i].a<<" "<<s1[i].b<<" "<<s1[i].c<<endl;
    cout<<"Finito"<<endl ;
}
```

ed assumendo che nel file di nome **dati** siano memorizzati 4 oggetti di tipo **s**

- a) il programma stampa il contenuto dei primi 2 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 che può causarne l'interruzione prima dell'esecuzione dell'ultima istruzione
- d) nessuna delle altre risposte è corretta

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

```
int c ;
void inizializza(int i){c = i ;}
int prossimo() { return c++ ; }
```

```
main() {
    int i, c ; cin>>i ;
    inizializza(i) ;
    c = 2 ; int d = prossimo() ;
    c = 1 ; int e = prossimo() ;
    if (d < e) cout<<"Minore" ;
}
```

- a) per qualsiasi numero intero immesso dall'utente, stampa sempre Minore
- b) contiene un errore di gestione della memoria
- c) nessuna delle altre risposte è vera
- d) il programma si interrompe in un qualche punto prima dell'istruzione condizionale

3. (2, -0.5)

```
for (int i=9; i>0; i++) {  
    cout<<i<<endl ;  
    i -= 3 ;  
}
```

è un ciclo che:

- a) produce un errore a tempo di esecuzione
- b) stampa i numeri dispari da 9 a 1 (estremi inclusi)
- c) a meno di problemi di overflow, va avanti all'infinito
- d) non produce nessuno dei risultati riportati nelle altre risposte

4. (2, -0.5) Il seguente frammento di codice:

```
char c;  
cin>>c;  
if (static_cast<int>(c) > 60)  
    cout<<"Maggiore" ;
```

- a) stampa o meno **Maggiore** a seconda del valore della costante carattere 'c'
- b) stampa **Maggiore** per ogni valore di **c** maggiore del valore 60
- c) contiene un errore di gestione della memoria
- d) nessuna delle altre risposte è corretta

5. (2, -1) Dato il seguente programma:

```
int fun(const int a[], int N)  
{  
    int b, i ;  
    for(i = 0 ; i < N ; i++) b += a[i] ;  
    return b ;  
}  
main()  
{  
    int c[] = {2, 4, 1, 8, 9} ;  
    cout<<fun(c, 3) ;  
}
```

- a) Il programma genera un errore a tempo di compilazione perché la variabile **N** attesa come secondo parametro dalla funzione **fun** non è mai definita nel resto del programma
- b) Il programma contiene un errore di gestione della memoria
- c) Nessuna delle altre risposte è vera
- d) Il programma stampa correttamente la somma dei primi 3 elementi dell'array **c**

6. (3, -0.5) Il seguente programma:

```
double a = 3 ;  
  
int fun1(const double &b, int n)  
{  
    a++ ;  
    n -= static_cast<int>(b) ;  
    return n ;  
}  
  
void fun2(double &a)  
{  
    fun1(a, 5) ;  
}
```

```

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

```

- contiene un errore di gestione della memoria
- stampa 5 3
- stampa 3 -1
- nessuna delle altre risposte 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

7. Supponendo che le operazioni di apertura (sia in scrittura che lettura) e scrittura su file non falliscano, il seguente programma

```

void fun1(const char a[], int n){ ofstream f("nome");
                                f.write(a, sizeof(char) * n); }
void fun2(char a[], int n){ ifstream f("nome");
                            for (int i = 0 ; i < n ; i++)
                                if (! (f>>a[i]) ) break ; }

```

```

main() {
    const int N = 3 ;
    char b[N] = {'a', 'b', 'c'}, c[] = {'d', 'd', 'd'};
    fun1(b, 3) ;
    fun2(c, 3) ;
    for (int i = 0 ; i < N ; i++) cout<<c[i] ; }

```

- memorizza il carattere 'a' nel primo byte del file di testo **nome**
  - stampa **abc**
  - stampa qualcosa di diverso da **abc**
  - non memorizza correttamente in **c** i caratteri contenuti in **b** mediante la funzione **fun2**, perché **b** è stato precedentemente memorizzato in forma binaria nel file **nome**
8. Due algoritmi equivalenti:
- Se implementati danno luogo allo stesso programma
  - Prevedono gli stessi passi
  - A parità di dati di ingresso possono portare a risultati diversi
  - A parità di dati in ingresso possono avere tempi di esecuzione diversi

9. Dato il seguente programma

```

main() {
    int i = 0 ; cin>>i ;
    switch(i) {
        case 'b':
            cout<<"Primo" ;
        case 'c':
            cout<<"Secondo" ;
            break ;
        default: cout<<"Errore" ;
    }
}

```

- a) Se l'utente immette il codice numerico usato per rappresentare il carattere **c**, stampa **Secondo**
- b) Se l'utente immette il codice numerico usato per rappresentare il carattere **b**, stampa **Primo**
- c) Se l'utente immette il carattere **b** stampa **Errore**
- d) Se l'utente immette il codice numerico usato per rappresentare il carattere **b**, stampa **Errore**

10. Dato il seguente programma

```
int g = 1 ;

int fun(char &g)
{
    return 2 * (++g);
}
main()
{
    char m = 1 ; char n = static_cast<int>(2.1) + fun(m) ;
    cout<<static_cast<int>(m * n) ;
}
```

- a) Se eseguito, il programma stampa **12**
- b) Nella inizializzazione della variabile **n**, la conversione **static\_cast<int>(2.1)** causa perdita di informazione
- c) Nell'ultima istruzione del **main**, la conversione esplicita non causa alcuna perdita di informazione
- d) Nessuna delle altre risposte è vera

### 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

11. (5 pt.) Cosa stampano a video le istruzioni di uscita del seguente codice, supponendo che l'utente scriva **esatto** seguito da INVIO e poi **pres\*tissimo** seguito di nuovo da INVIO?

```
main()
{
    const int N = 100 ; char stringa[N] ;
    cin>>stringa ;
    cout<<stringa<<endl ;
    stringa[0] = '\\0' ;
    int i = 0 ;
    do {cin>>stringa[i] ; i++; }
        while (i <= N && stringa[i-1] != '*') ;
    cout<<stringa<<endl ;
}
```

12. (6 pt) Scrivere una funzione che prenda in ingresso un vettore di  $n$  elementi interi, un indice  $i$  ed un numero intero positivo  $m$ . Se non sono presenti valori negativi tra la posizione di indice  $i$  e quella di indice  $i + m$ , la funzione sposta l'elemento di indice  $i$  di  $m$  posizioni in avanti, scambiandolo con quello di indice  $i+m$ . Altrimenti scambia l'elemento di indice  $i$  con uno degli elementi di valore negativo ed indice compreso tra  $i$  ed  $i+m$ . Ad esempio, se si passa alla funzione il vettore  $[1, 4, 2, 5, 6, 9, 3]$ , ed i valori  $i=1$  ed  $m=5$ , si ottiene  $[1, 9, 2, 5, 6, 4, 3]$ . Se invece si passa il vettore  $[1, 4, 2, 5, -1, 9, 3]$ , ed i valori  $i=1$  ed  $m=5$ , si ottiene  $[1, -1, 2, 5, 4, 9, 3]$ . Gestire opportunamente le situazioni di errore.



**Programmazione I / Informatica generale**  
**Prova scritta**  
**29 Gennaio 2008**

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

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

**Avete due copie di questa pagina, usatene una per calcolare il voto da soli dopo la correzione. Indicare le risposte corrette apponendo una croce nella casella corrispondente.**

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					2	-1
2					3	-0.5
3					2	-0.5
4					2	-0.5
5					2	-1
6					3	-1
7						
8						
9						
10						

**Risposta alla domanda 11:**

**Risposta alla domanda 12:**



**Programmazione I / Informatica generale**  
**Prova scritta**  
**29 Gennaio 2008**

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

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

**Avete due copie di questa pagina, usatene una per calcolare il voto da soli dopo la correzione. Indicare le risposte corrette apponendo una croce nella casella corrispondente.**

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					2	-1
2					3	-0.5
3					2	-0.5
4					2	-0.5
5					2	-1
6					3	-1
7						
8						
9						
10						

**Risposta alla domanda 11:**

**Risposta alla domanda 12:**