

# Programmazione I / Informatica generale

## Prova scritta - 30 giugno 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, -.5) Il seguente programma:

```
int a = 3 ;

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

main()
{
    double b = 3.0 ; int c = 3 ;
    b = fun1(b, c) ;
    fun1(b, c) ;
    cout<<a<<" "<<static_cast<int>(b)<<" "<<c ;
}
```

- a) contiene un errore di gestione della memoria
- b) stampa 5 3 3
- c) stampa 5 0 3
- d) nessuna delle altre risposte è vera

2. (3, -.5) Il seguente programma:

```
int a = 3 ;
void fun(int &a, char c){ a += c ; }
main()
{
    int c = 10 ;
    fun(c, static_cast<char>(a)) ;
    cout<<a<<" "<<c ;
}
```

- a) Stampa: 3 13
- b) Contiene un errore di gestione della memoria
- c) Stampa 6 10
- d) Nessuna delle precedenti risposte è vera

3. (2, -.5) Dato un file sorgente C/C++ contenente la seguente direttiva

```
#define COSTANTE1 15
```

- a) tale direttiva è perfettamente equivalente all'istruzione  
`const int COSTANTE1 = 15 ;`
- b) il preprocessore sostituisce testualmente ogni occorrenza della parola **COSTANTE1** con la seguente sequenza di due caratteri: **15**
- c) la parola **COSTANTE1** è ancora presente nel file prodotto dal preprocessore e preso in ingresso per la traduzione
- d) la presenza di tale direttiva e della seguente istruzione  
`int COSTANTE1 ;`  
all'interno di una qualche funzione definita in un punto successivo del file (successivo rispetto a quello in cui è presente la direttiva) non comporta errori di compilazione

4. (1, -.5) In C/C++, una stringa implementata mediante array di caratteri:

- a) va sempre allocata dinamicamente;
- b) può essere copiata in un'altra stringa con un assegnamento del tipo `stringa2=stringa1;`
- c) nessuna delle altre risposte è vera;
- d) deve sempre essere terminata dal carattere di fine stringa `'\0'`.

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

```
main(){double c, d ; cin>>c>>d ; cout<<( (c/d)*d == c ) ;}
```

- a) il programma contiene un errore che ne impedisce la compilazione
- b) il programma stampa **1** qualsiasi siano i valori dei due numeri inseriti dall'utente su *stdin*
- c) il programma stampa **0** qualsiasi siano i valori dei due numeri inseriti dall'utente su *stdin*
- d) nessuna delle altre risposte è vera

6. (+2, -.5) Data la seguente definizione:

```
char c = static_cast<char>(static_cast<int>('z')) ;
```

- a) si ha perdita di informazione in almeno una delle conversioni
- b) subito prima dell'assegnamento viene effettuata una conversione implicita
- c) se si tentasse di stampare il valore di **c** mediante `cout<<c;` si stamperebbe il codice ASCII del carattere **'z'**
- d) nessuna delle altre risposte è corretta

## **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. Dato un programma scritto in linguaggio C/C++

- a) Quando inizia l'esecuzione del programma viene allocato in memoria il record di attivazione della funzione **main**
- b) Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili globali a cui accede la funzione
- c) L'ordine in cui saranno allocati i record di attivazione delle funzioni non è in generale noto a tempo di scrittura di un programma
- d) Quando una funzione ne invoca un'altra durante l'esecuzione di un programma, il record di attivazione della funzione chiamata viene allocato subito sopra quello della funzione chiamante

8. Data l'istruzione `cin>>i` ; ove `i` è una variabile di tipo `int` precedentemente definita:
- Se, all'esecuzione di tale istruzione, lo `stdin` già contiene una sequenza di caratteri che rappresenta un numero intero, il programma non si blocca in attesa dell'utente, ma prova immediatamente a memorizzare tale numero nella variabile `i`
  - All'esecuzione di tale istruzione il programma si blocca sempre in attesa che l'utente invii dei caratteri sullo `stdin`
  - Se, all'esecuzione di tale istruzione, lo `stdin` contiene una sequenza di caratteri che rappresenta un numero intero, seguita da uno spazio ed infine da una sequenza di caratteri che non rappresenta alcun numero, il `cin` non va in stato di errore
  - Nel caso in cui tale istruzione sia eseguita con successo, ossia venga memorizzato correttamente un numero intero letto da `stdin` nella variabile `i`, lo `stdin` è svuotato di tutto il suo contenuto

9. Dato il seguente programma:

```
main()
{
    ofstream f("nome.txt") ;
    f<<1<<" "<<2<<endl ;
    f<<3 ;
    f.close() ;
    ifstream f2("nome.txt") ;
    int i ;
    while(f2>>i)
        cout<<i<<" " ;
}
```

- al termine del programma, il file `nome.txt` è un file binario contenente i numeri 1, 2 e 3
- al termine del programma, il file `nome.txt` è un file di testo contenente i numeri 1, 2 e 3
- il programma stampa `1, 2, 3,`
- anche in assenza di errori di scrittura, quando il file di testo `nome.txt` viene riaperto in lettura con l'istruzione `ifstream f2("nome.txt")` ; può non contenere tutti i numeri che sono stati precedentemente immessi nell'`ofstream f`

### 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 scrive su `stdout` il seguente programma?

```
void fun(char s[], int n)
{
    int i = 0, j = n - 1 ;
    char c = s[0] ;
    while (j != 0){
        s[j] = c ;
        i++ ; j-- ;
        c = s[i] ;
    }
}
```

```
main()
{
    char string[] = "esca";
    cout<<string<<endl;
    fun(string, strlen(string)); // strlen ritorna la lunghezza della
                                // stringa passata come argomento
    cout<<string<<endl;
}
```

11. **(6 pt)** Scrivere una funzione che prende in ingresso una stringa e ritorna una nuova stringa in cui ogni sequenza di caratteri uguali contenuta nella stringa in ingresso è stata sostituita da una sola occorrenza di tale carattere. Ad esempio, presa in ingresso la stringa *"aabhaggggh"*, la funzione deve ritornare la stringa *"abhagh"*. L'*array* di caratteri in cui è memorizzata la stringa ritornata deve avere le dimensioni minime necessarie per contenere tale stringa.



# Programmazione I / Informatica generale

## Prova scritta - 30 giugno 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					1	-0,5
5					2	-0,5
6					2	0-,5
7						
8						
9						

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

**Risposta alla domanda 11 (6 pt):**



# Programmazione I / Informatica generale

## Prova scritta - 30 giugno 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					1	-0,5
5					2	-0,5
6					2	0-,5
7						
8						
9						

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

**Risposta alla domanda 11 (6 pt):**