

# Programmazione I / Informatica generale

## Prova scritta

### 11 Giugno 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 ;
```

Infine, quando 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() {
    for (unsigned int i = 5 ; i >= 0 ; i--)
        cout<<i<<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

2. (3, -0.5) Data la seguente funzione e supponendo che: 1) l'array **a** abbia dimensione maggiore o uguale di **num\_elem+1**, sia correttamente allocato in memoria e sia utilizzato per rappresentare un vettore di **num\_elem** interi nel momento in cui la funzione viene chiamata, 2) il valore di **pos** sia compreso tra 0 e **num\_elem-1**, la funzione

```
void fun(int a[], int &num_elem, const int pos, const int v)
{
    for (int i = num_elem ; i >= pos ; i--)
        a[i+1] = a[i] ;
    a[pos] = v ;
    num_elem++ ;
}
```

- a) inserisce correttamente il valore **v** nella posizione **pos** nel vettore rappresentato mediante l'array **a**
- b) sovrascrive tutti gli elementi di indice compreso tra **pos** e **num\_elem** con il valore **v**
- c) contiene un errore di gestione della memoria
- d) nessuna delle altre risposte è vera

3. (4 pt, -0.5) Dato il seguente programma:

```
int *a, num_elem ;
void distruggi() { delete [] a; num_elem = 0 ; }
void inserisci(int n) { a[num_elem] = n ; num_elem++ ; }
void stampa() { for (int i = 0 ; i < num_elem ; i++) cout<<a[i]<<" " ; }
```

```
main() {
    a = new int[10] ;
    inserisci(1) ; stampa() ;
    distruggi() ;
    int *b = new int[5] ; b[0] = 3 ;
    inserisci(4) ; stampa() ; cout<<b[0]<<endl ;
}
```

- a) il programma stampa 1 4 3
- b) nessuna delle altre risposte è vera
- c) il programma contiene un errore di *memory leak*
- d) il programma contiene un errore di gestione della memoria diverso dal *memory leak*

4. (3, -0.5) Dato il seguente programma:

```
struct ts {char s[10] ; int n ;} ;
```

```
main() {
    ts a ;
    strcpy(a.s, "prova") ; // memorizzazione della stringa "prova" nel
                          // campo a.s
    a.n = 10 ;
    ts b ; b = a ;
    cout<<b.s<<" "<<b.n ;
}
```

- a) l'istruzione di assegnamento  $b=a$  ; da luogo ad un errore a tempo di compilazione
- b) il programma contiene un errore di gestione della memoria
- c) il programma stampa **prova 10**
- d) nessuna delle altre risposte è vera

5. (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 da *stdin* **2 prova** (c'è uno spazio tra il carattere 2 e la stringa prova) e poi preme INVIO, il programma stampa **2 prova**
- b) se l'utente inserisce da *stdin* **2 prova** (c'è uno spazio tra il carattere 2 e la stringa prova) e poi preme INVIO, il programma memorizza il valore 2 nella variabile *i*, e poi si blocca sull'istruzione  $cin>>s$  ; finché l'utente non inserisce dei nuovi caratteri e preme INVIO
- c) se l'utente inserisce da *stdin* **2 prova** (c'è uno spazio tra il carattere 2 e la stringa prova) e poi preme INVIO, il *cin* va in stato di errore sull'istruzione  $cin>>i$  ;
- 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

6. Dato il seguente programma:

```
main()
{
    int i, a[3] = {0, 1, 0} ;
    cin>>i ;
    if (a[i] != 0 && (i >= 0 && i < 3)) {
        a[i] = 10 ;
        cout<<a[i] ;
    }
}
```

- a) Se l'utente immette 1 su *stdin* il programma stampa 10
- b) Il programma contiene un errore logico
- c) Il programma contiene un errore di accesso alla memoria
- d) Se l'utente immette 0 su *stdin* il programma stampa 0

7. Dato il seguente programma:

```
void inizializza(int a[][3], int N)
{
    for(int i = 0 ; i < N ; i++)
        for(int j = 0 ; j < 3 ; j++)
            a[i][j] = 1 ;
}
main()
{
    int c[2][3] ;
    inizializza(c, 2) ;
    // seguono istruzioni per la stampa della matrice ...
}
```

- a) La matrice *c* è vuota, ossia i suoi elementi non hanno nessun valore, quando viene passata alla funzione *inizializza*, che poi provvederà a dare un valore a ciascun elemento
- b) In quanto all'accesso agli elementi della matrice *a* all'interno della funzione *inizializza*, conoscere il numero di colonne della matrice serve solo per evitare accessi fuori dalla matrice stessa
- c) In quanto all'accesso agli elementi della matrice *a* all'interno della funzione *inizializza*, conoscere il numero di righe della matrice serve solo per evitare accessi fuori dalla matrice stessa
- d) Dopo l'invocazione della funzione *inizializza*, tutti gli elementi della matrice *c* contengono il valore 1

8. Dato il seguente programma:

```
void scrivi(ofstream &f) { f<<"prova" ; f.close() ; }
main()
{
  ofstream of("dati.txt") ;
  scrivi(of) ;
  ifstream is("dati.txt") ;
  char buf[256] ;
  is>>buf ;
  cout<<buf ;
}
```

a) il programma stampa **prova** su *stdout*

b) quando si riapre il file *dati.txt* in lettura all'interno della funzione *main* non è garantito che il suo contenuto sia stato aggiornato

c) quando si riapre il file *dati.txt* in lettura, quest'ultimo contiene una sequenza di byte, ciascuno dei quali contiene a sua volta il codice di uno dei caratteri precedentemente scritti nel file stesso mediante la funzione *scrivi*

d) anche se il parametro formale *f* della funzione *scrivi* muore in uscita dalla funzione stessa, l'invocazione della funzione *f.close()* nella funzione *scrivi* è comunque necessaria per provocare l'aggiornamento del file *dati.txt*

### 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

9. (4 pt.) Cosa scrive su *stdout* il seguente programma?

```
void fun(char s[], int n, char c)
{
  for (int i = 0 ; s[i] != '\0' ; i++) {
    if (s[i] == c) {
      if(s[i+1] != '\0')
        s[i+1] = s[i];
    }
    else if (i != 0)
      s[i] = s[i+1] ;
  }
}

main()
{
  char string[] = "ikea";
  cout<<string<<endl ;
  fun(string, strlen(string), 'e');
  cout<<string;
}
```

10. (5 pt.) Scrivere il corpo della seguente funzione

```
int *fun(int a[], int N, int v) ;
```

in cui il parametro  $N$  è assunto contenere il numero di elementi dell'array  $a$ , e che ritorna un array dinamico, anch'esso di dimensione  $N$ , contenente tutti gli elementi dell'array  $a$  di valore minore o uguale a  $v$ , seguiti da tutti gli elementi dell'array  $a$  di valore maggiore di  $v$ .



# Programmazione I / Informatica generale

## Prova scritta - 11 Giugno 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/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					4	-0.5
4					3	-0.5
5					3	-0.5
6						
7						
8						

**Risposta alla domanda 9 (4 pt):**

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



# Programmazione I / Informatica generale

## Prova scritta - 11 Giugno 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/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					4	-0.5
4					3	-0.5
5					3	-0.5
6						
7						
8						

**Risposta alla domanda 9 (4 pt):**

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