

Programmazione I

Prova scritta - 13 giugno 2011 - 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. (2, -0.5) 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

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

```
void fun1(const char a[], int n){ ofstream f("nome_file");
        f.write(a, sizeof(char) * n); }
void fun2(char a[], int n){ ifstream f("nome_file");
        for (int i = 0 ; i < n ; i++)
            if (! (f>>a[i]) ) break ; }
main() {
    char b[] = {'a', 'b', 'c'}, c[] = {'d', 'd', 'd'};
    fun1(b, 3) ;
    fun2(c, 3) ;
    for (int i = 0 ; i < 3 ; i++) cout<<c[i] ;
}
```

- a) crea un file di testo contenente i caratteri **abc**
- b) stampa qualcosa di diverso da **abc**
- c) 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_file**
- d) nessuna delle altre risposte è vera

3. (4, -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 (con **num_elem > 0**) nel momento in cui la funzione viene chiamata, 2) il valore di **pos** sia compreso tra 0 e **num_elem**, la funzione

```

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

```

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

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

```
int g = 1 ;
```

```

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

```

```

main()
{
    const int m = 1 ;

```

```

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

```

- Se eseguito, il programma stampa 9
- Se eseguito, il programma stampa 4
- Nell'ultima istruzione del **main**, la conversione esplicita non causa alcuna perdita di informazione
- Nesuna 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

5. Indicare quali delle seguenti affermazioni sono vere

- E' tipicamente meglio utilizzare un linguaggio compilato per implementare algoritmi che è importante eseguire molto velocemente
- Il linguaggio macchina permette di scrivere programmi portabili tra diverse architetture
- Il linguaggio macchina non è un linguaggio di alto livello
- Un programma in linguaggio macchina è una sequenza di byte da non interpretarsi come sequenza di caratteri

6. Dato il seguente programma:

```
1:float b = 3.5;
2:float fun(float &a)
3:{
4:    int i ;
5:    for (i = 0 ; i < 2 ; i++)
6:        a *= 2 ;
7:    return a + b + i ;
8:}
9:
10:main()
11:{
12:    float b = 1.5 ;
13:    float c = fun(b) ;
14:    cout<<static_cast<int>(c + b)<<endl ;
15:}
```

- a) quando invocata alla riga 13, la funzione **fun** modifica il valore della variabile **b** definita alla riga 12
- b) il parametro formale **a** definito alla riga 2 non è visibile alla riga 14
- c) la variabile **i** definita alla riga 4 non è visibile alla riga 7
- d) il programma stampa 17

7. La seguente istruzione:

```
cout<<static_cast<char>('b' - 'a' + '3') ;
```

- a) può causare problemi di *overflow*
- b) immette il codice di un carattere stampabile sullo *stream* di uscita standard (**cout**)
- c) stampa 4
- d) nessuna delle altre risposte è corretta

8. Un oggetto allocato dinamicamente all'interno di una funzione,

- a) occupa memoria dall'inizio alla fine del programma;
- b) è effettivamente allocato solo quando l'istruzione di allocazione viene eseguita;
- c) smette di occupare spazio in memoria quando viene deallocato;
- d) se non esplicitamente deallocato prima, è implicitamente deallocato al termine della funzione.

9. Dato il seguente programma:

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

- a) Se l'utente immette 1 su *stdin* il programma stampa 1
- 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

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

```
struct s {int a; char b[2]; } ;

s &fun(s &a)
{
    a.a = 2 ;
    a.b[0] = 'c' ;

    return a ;
}

main()
{
    s b = {0, {'r', '\0' }} ;
    s &c = fun(b) ;

    cout<<b.b<<" "<<c.b<<endl ;
}
```

11. **(5 pt)** Scrivere una funzione che prenda un ingresso un vettore di numeri interi v , e, supponendo che v contenga valori tutti diversi tra loro, lo modifichi eliminandone l'elemento di valore minimo m e quello di valore massimo M . La funzione ritorni infine un nuovo vettore contenente m ed M . Ad esempio, la funzione trasformerebbe il vettore $[3, 7, 6, 2, 5]$ in $[3, 6, 5]$ e ritornerebbe $[2, 7]$. Se ritenuto opportuno definire ed utilizzare ulteriori funzioni di supporto.

Programmazione I

Prova scritta - 13 giugno 2011

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è la copia da tenere per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					2	-0,5
2					3	-0,5
3					4	-0,5
4					3	-0,5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (3 pt):

Risposta alla domanda 11 (5 pt):

Programmazione I

Prova scritta - 13 giugno 2011

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è la copia da tenere per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					2	-0,5
2					3	-0,5
3					4	-0,5
4					3	-0,5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (3 pt):

Risposta alla domanda 11 (5 pt):