

Programmazione I

Prova scritta - 5 settembre 2012 - 1h20min

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 valutata 0

1. (3, -.5) Tenendo presente eventuali problemi di conversione e di rappresentabilità, il seguente frammento di codice

```
int delta_w;  
unsigned int wsum;  
cin>>delta_w>>wsum;  
if (-delta_w > wsum)  
    cout<<"maggiore"<<endl ;
```

- a) Stampa maggiore solo se **delta_w** è negativo e l'espressione algebrica **-delta_w > wsum** è vera
- b) Stampa maggiore se **delta_w** è negativo e **wsum** contiene un valore negativo
- c) **Può stampare maggiore anche se delta_w è positivo**
- d) Nessuna delle altre risposte è vera

2. (3, -.5) Il seguente frammento di codice:

```
for (int i=5; i >0 ; i--) { if (i==3) i++; cout<<i*2<<" " ;}
```

- a) Stampa 10 8 6
- b) **Contiene un ciclo infinito**
- c) Nessuna delle altre risposte è vera
- d) Stampa i numeri pari 10 8 8 6 4 2

3. (3, -.5) Data la seguente stringa rappresentata mediante un array

```
char s[10] = "Soprano" ;
```

le istruzioni:

```
s[2] = 't' ; s[3] = s[2] ; s[4] = 'o' ; s[5] = '\\0' ;
```

- a) contengono uno o più errori di accesso alla memoria
- b) **trasformano la stringa da “Soprano” a “Sotto”**
- c) non modificano la lunghezza della stringa
- d) nessuna delle altre risposte è vera

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

**PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE -
Ogni domanda può avere da una a quattro risposte CORRETTE.**

- Ogni risposta esatta viene calcolata: +1
 - Ogni risposta errata viene calcolata: -0.5
 - Una risposta lasciata in bianco viene calcolata: 0
5. Data una sequenza di N valori interi positivi
- a) La struttura dati più efficiente, in termini di occupazione di memoria, in cui si potrebbero memorizzare tali valori è un array di N elementi
 - b) Aggiungere un elemento nel mezzo della sequenza ha costo $O(N)$ se la sequenza è implementata mediante un array
 - c) Accedere all' i -esimo elemento della sequenza ha costo $O(1)$ se la sequenza è implementata mediante una lista semplice
 - d) Se la sequenza è memorizzata in un *array* utilizzando un terminatore per delimitare la fine della sequenza stessa, non è necessario mantenere ulteriori informazioni per poter accedere sequenzialmente a tutti gli elementi della sequenza

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, c = fun(b) ;
13:    cout<<static_cast<int>(c)<<endl ;
14:}
```

- a) la funzione **fun** non modifica il valore della variabile **b** definita alla riga 12
- b) il parametro formale **a** definito alla riga 2 non è visibile alla riga 13

c) la variabile **i** definita alla riga 4 non è visibile alla riga 7

d) **il programma stampa 9**

7. Dato il seguente programma:

```
int fun(int b[], int n)
{
    int s = 0 ;
    for (int i = 0 ; i < n ; i++) {      s += b[i] ; }
    delete [] b ;
    return s ;
}
```

```
main()
{
    int *a = new int[2] ; a[0] = 2 ; a[1] = 4 ;

    cout<<fun(a, 2)<<" " ;
    for (int i = 0 ; i < 2 ; i++)
        cout<<a[i]<<" " ;
}
```

a) il programma stampa **6 2 4**

b) **nella funzione fun, all'interno del ciclo for non si modifica l'array dinamico allocato con l'operatore new all'inizio della funzione main**

c) **quando l'esecuzione arriva all'inizio del ciclo for nella funzione main, il puntatore a non punta ad un oggetto correttamente allocato in memoria**

d) **il programma contiene un errore di gestione della memoria**

8. 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 causa perdita di informazione**

d) **Quando invocata nel main, la funzione fun non modifica la variabile globale g**

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

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

```
main()
{
    bool finito = false ;
    for( int i = 0 ; ! finito ; ) {
        cout<<i<<" " ;
        if (i == 2)
            finito = true ;
        else
            cout<<i<<" " ;
        i++ ;
    }
}
```

10. **(5 pt.)** Scrivere una funzione che prenda in ingresso una matrice di caratteri di dimensioni note, per la quale si assume che ogni riga contiene una stringa, e modifichi la matrice affinché la prima riga della matrice contenga la stringa S che precede tutte le altre in ordine alfabetico, ed il precedente contenuto della prima riga sia spostato nella riga che prima conteneva la stringa S .

Programmazione I

Prova scritta - 4 settembre 2012

Nome: _____ Cognome: _____

Matricola: _____

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 l'unica pagina che dovete consegnare. Per comodità avete anche un copia di questa pagina per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					3	-0.5
4					3	-0,5
5						
6						
7						
8						

Risposta alla domanda 9 (4 pt):

Risposta alla domanda 10 (5 pt):

Programmazione I

Prova scritta - 4 settembre 2012

Nome: _____ Cognome: _____

Matricola: _____

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 l'unica pagina che dovete consegnare. Per comodità avete anche un copia di questa pagina per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					3	-0.5
4					3	-0,5
5						
6						
7						
8						

Risposta alla domanda 9 (4 pt):

Risposta alla domanda 10 (5 pt):