

Programmazione I

Prova scritta - 13 febbraio 2013 - 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, -0.5) Dato il seguente programma:

```
main()
{
    int i = 1 , a[3] = {0} ;
    do {
        cin>>a[i] ;
        if (a[i] != a[i-1] + 1)
            i-- ;
        else
            i++ ;
    } while(i < 3) ;
    for(int j = 0 ; j < 3 ; j++)
        cout<<a[j]<<" " ;
}
```

- a) Se eseguito per intero il programma stampa 0 1 2
- b) Il programma viene terminato forzatamente prima dell'inizio del ciclo **for** finale
- c) Nessuna delle altre risposte è vera
- d) Il programma contiene errori di gestione della memoria

2. (3, -.5) Data la seguente stringa rappresentata mediante un array

```
char s[] = "abcdef" ;
```

le istruzioni:

```
s[2] = s[5] = '\0' ;
```

- a) contengono uno o più errori di accesso alla memoria
- b) trasformano la stringa originaria in due stringhe di lunghezza 2
- c) non modificano la lunghezza della stringa
- d) nessuna delle altre risposte è vera

3. (2, -0.5) Dato il seguente programma e supponendo il codice del carattere a sia 97

```
main() {
    char a = 'a'; int b = 3 ;
    cout<<(a + b) ;
}
```

- Il programma stampa 100
- Il programma stampa d
- Il programma genera un errore a tempo di compilazione perché non si può effettuare la somma tra un valore di tipo *char* ed uno di tipo *int*
- 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

4. Dato il seguente programma:

```
main()
{
    enum stato {pieno, vuoto} ;
    stato v[3] = {pieno, pieno, vuoto} ; int i ;
    cin>>i ;
    if ( ( i >= 0 && i < 3 ) && v[i] != vuoto ) {
        v[i] = vuoto ;
        cout<<"Svuotato" ;
    }
}
```

- Il programma contiene un errore logico
- Il programma contiene un errore di accesso alla memoria
- Se l'utente immette 1 su *stdin* il programma stampa **Svuotato**
- Se l'utente immette 3 su *stdin* il programma non stampa nulla e termina correttamente

5. Nel seguente programma:

```
1:int a = 1;
2:
3:fun()
4:{
5:    int i ;
6:    for (i = 0 ; i < 5 ; i++) {
7:        double a = i / 2.0 ;
8:        cout<<a * i ;
9:    }
10:    cout<<a + i ;
11:}
12:
13:main()
14:{
15:    double b = 1.4 ;
16:    fun() ;
17:    cout<<a + b<<endl ;
18:}
```

- a) la variabile **b** definita alla riga 15 non ha visibilità pari all'intero programma;
- b) la variabile **a** definita alla riga 1 ha tempo di vita pari all'intero programma;
- c) la variabile **b** definita alla riga 15 ha tempo di vita pari all'intero programma;
- d) la variabile **a** definita alla riga 1 non è visibile alla riga 8.

6. Dato il seguente programma:

```
const int N = 10 ; struct ss { char a[N] ; } ;
struct ss fun(ss & e) { e.a[0] = 'z' ; return e ;}
main()
{
    ss c, d ; c.a[0] = 'y' ;
    d = fun(c) ; d.a[0] = 'k' ; cout<<c.a[0] ;
}
```

- a) Il programma stampa **k**
 - b) Il programma stampa **z**
 - c) Il ritorno di **e** nella funzione **fun** comporta la copia del contenuto del parametro attuale a cui si riferisce il parametro formale **e**
 - d) Nessuna delle altre risposte è vera
7. Data l'istruzione **f>>i** ; ove **i** è una variabile di tipo **int** ed **f** è un oggetto di tipo **ifstream**, configurato per leggere numeri in base 10 ed associato ad un file correttamente aperto:
- a) Se, all'esecuzione di tale istruzione, non si è ancora raggiunta la fine del file, il programma legge i byte del file a partire dalla posizione corrente, alla ricerca della rappresentazione in base 10 di un numero intero
 - b) All'esecuzione di tale istruzione il programma si blocca sempre in attesa che vengano aggiunti nuovi caratteri nel file associato all'*ifstream* **f**
 - c) Se, all'esecuzione di tale istruzione, la porzione di file a partire dalla posizione corrente 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, l'oggetto **f** non va in stato di errore
 - d) Nel caso in cui tale istruzione sia eseguita con successo, ossia venga memorizzato correttamente un numero intero letto da **f** nella variabile **i**, eventuali caratteri successivi a quelli che rappresentano il numero letto vengono lasciati nell'*ifstream*
8. Quali delle seguenti affermazioni sono vere?
- a) La funzione **printf** della libreria standard del C ha un numero di argomenti variabile
 - b) Nella funzione **scanf** della libreria standard del C, sbagliare il tipo o l'indirizzo della variabile in cui si desidera scrivere il valore letto da *stdin* può causare corruzione della memoria
 - c) La funzione **printf** della libreria standard del C determina automaticamente il formato con cui stampare ciascuno oggetto passato come argomento (dopo la stringa *format*) in base al tipo dell'oggetto stesso
 - d) In C non esiste il passaggio di un oggetto per riferimento, ma si può emularlo passando l'indirizzo dell'oggetto stesso

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. **(5 pt.)** Descrivere sintassi e semantica dell'istruzione iterativa **for** in non più di sei righe e senza usare diagrammi.

10. (6 pt) Scrivere una funzione che prende in ingresso una stringa contenente solo lettere minuscole, e ritorna una nuova stringa contenente solo i caratteri presenti una sola volta nella stringa in ingresso. Ad esempio, presa in ingresso la stringa “*rinoceronte*”, la funzione ritornerebbe la stringa “*ict*”. Cercate di realizzare una funzione con costo computazionale più basso possibile.

Programmazione I

Prova scritta - 13 febbraio 2013

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					2	-0,5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (5 pt):

Risposta alla domanda 10 (6 pt):

Programmazione I

Prova scritta - 13 febbraio 2013

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. Usate questa copia 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					2	-0,5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (5 pt):

Risposta alla domanda 10 (6 pt):