

Programmazione I

Prova scritta - 05 febbraio 2014 - 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) Il seguente programma:

```
struct pers {char nome[25]; int anno;} ;
pers fun(pers p, const char *s)
{ strcpy(p.nome, s) ; p.anno = 2000; return p ; }
main()
{
  pers z = {"luigi", 1985};
  pers y = fun(z, "marco") ;
  cout<<z.nome<<" "<<y.nome ;
  strcpy(z.nome, "anna") ; cout<<" "<<y.anno<<" "<<y.nome ;
}
```

- a) Stampa **marco marco 2000 anna**
- b) Stampa **luigi marco 1985 marco**
- c) Stampa **marco marco 2000 marco**
- d) Nessuna delle altre risposte è corretta

2. (2, -0.5) Dato il seguente programma e supponendo il codice del carattere *b* sia 98, nonché facendo attenzione alle conversioni implicite:

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

- a) Il programma stampa **101**
- b) Il programma stampa **e**
- c) Il programma genera un errore a tempo di compilazione perché non si può effettuare la somma di un valore di tipo *int* con un valore di tipo *char*
- d) Nessuna delle altre risposte è vera

3. (2, -0.5) Date le istruzioni `cout<<'1'<<'2'<<endl`; e `cout<<12<<endl`; e supponendo che l'operatore di uscita sia configurato per stampare i numeri interi in base 10.
- Non è vero che entrambe le istruzioni immettono la stessa sequenza di codici carattere sullo *stdout*
 - Tali istruzioni immettono sullo *stdout* la stessa sequenza di codici carattere dell'istruzione `cout<<1<<2<<endl` ;
 - Nessuna delle altre risposte è vera
 - La seconda istruzione immette su *stdout* un numero maggiore di byte rispetto alla prima
4. (2, -0.5) Dato il seguente frammento di codice:
- ```
int a[3] ; delete [] a; a[0] = 3;
```
- il frammento non contiene errori di gestione della memoria;
  - l'assegnamento `a[0] = 3`; comporta un errore di accesso alla memoria perché la memoria dinamica occupata da `a` è stata deallocata;
  - l'oggetto `a` non appartiene alla classe di memorizzazione a cui può essere applicato l'operatore `delete`
  - nessuna delle altre affermazioni 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. Due algoritmi equivalenti:
- Forniscono lo stesso risultato per qualsiasi insieme di dati in ingresso
  - Hanno lo stesso codice
  - Prevedono gli stessi passi
  - A parità di dati in ingresso possono avere tempi di esecuzione diversi
6. Dato il seguente programma:
- ```
int g = 20 ;

bool fun(int a) {
    if (a < 10) { ++g ; return true ;}
    return false ; }

main() {
    const int m = 22 ;

    if ((m % 2 == 0) || fun(m)) cout<<-g ;
    else cout<<g ;
}
```
- Se eseguito, il programma stampa -20
 - Se eseguito, il programma stampa -21
 - A parte l'inizializzazione, il valore della variabile globale `g` non è mai modificato
 - Durante l'esecuzione del programma, la funzione `fun` non è invocata all'interno del `main`

7. Dato il seguente programma e supponendo che non sorga alcun problema di approssimazione dovuto alla precisione limitata del tipo **double**:

```
1: double a = 3.1 ;  
2:  
3: void fun(int a, int b)  
4: { a += b ; }  
5: main()  
6: { fun(a, 3) ; cout<<a<<endl ; }
```

- a) la variabile **a** definita alla riga 1 ha scope relativo a tutto il programma;
- b) la variabile **a** definita alla riga 1 ha tempo di vita pari all'intero programma;
- c) prima dell'assegnamento, la variabile **a** riferita alla riga 4 ha un valore diverso da 3.1;
- d) la variabile **a** definita alla riga 1 è visibile alla riga 4

8. Indicare quali delle seguenti affermazioni sono vere

- a) A parità di algoritmo implementato, l'esecuzione di un programma scritto in un linguaggio compilato (e quindi poi compilato) è tipicamente più veloce dell'esecuzione di un programma scritto in un linguaggio interpretato
- b) Il linguaggio macchina permette di scrivere programmi portabili tra diverse architetture
- c) Il linguaggio macchina non è un linguaggio di alto livello
- d) Un programma in linguaggio macchina è una sequenza di byte da non interpretarsi come sequenza di caratteri

9. Quali delle seguenti affermazioni sono vere?

- a) In C non esiste il passaggio di un oggetto per riferimento, ma si può emularlo passando l'indirizzo dell'oggetto stesso
- b) In C non esiste il passaggio per riferimento e non c'è nessun modo per emularlo
- c) Dato un parametro formale di tipo puntatore ad un certo tipo **T**, allora, all'atto del passaggio dell'indirizzo di un oggetto di tipo **T** come parametro attuale nella posizione corrispondente a tale parametro formale, non avviene la copia del contenuto dell'oggetto di tipo **T** memorizzato a tale indirizzo
- d) Nessuna delle altre affermazioni è vera

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)** Descrivere sintassi e semantica dell'operatore logico **&&** in non più di sei righe, più un'eventuale tabella.

11. **(6 pt)** Senza utilizzare funzioni o oggetti di libreria per l'output formattato all'interno di stringhe, scrivere una funzione che prenda in ingresso un numero intero e crei e ritorni una nuova stringa contenente la rappresentazione in base 10 del numero, preceduta dal segno del numero. Se, ad esempio, viene passato il numero 34, al funzione ritorna la stringa "+34", mentre se viene passato il numero -278, la funzione ritorna la stringa "-278". Assumere che il numero passato non abbia più di 10 cifre. Si ottiene il punteggio massimo se l'*array* di caratteri ritornato ha le dimensioni minime necessarie per contenere la stringa.

Programmazione I

Prova scritta - 05 febbraio 2014

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					2	-0,5
3					2	-0,5
4					2	-0,5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (5 pt):

Risposta alla domanda 11 (6 pt):

Programmazione I

Prova scritta - 05 febbraio 2014

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					2	-0,5
3					2	-0,5
4					2	-0,5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (5 pt):

Risposta alla domanda 11 (6 pt):