

Programmazione I

Prova scritta - 11 Settembre 2019 - 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() {  
    char a = 60 ; char b = 3 ;  
    cout<<static_cast<char>(a+b) ;  
}
```

- a. Il programma stampa 63
- b. Il programma stampa il carattere di codice uguale a 63
- c. Il programma genera un errore a tempo di compilazione perché non si può effettuare la somma di due valori di tipo *char*
- d. Nessuna delle altre risposte è vera

2. (3, -0.5) Il seguente programma

```
main()  
{  
    for (int i = 1 ; i < 2 ; i++)  
        for (int j = i ; j < 3 ; j++) {  
            cout<<i<<" "<<j<<" " ;  
            if (j % 1 == 0)  
                j++ ;  
        }  
}
```

- a. stampa 1 1
- b. stampa 1 1 2 2
- c. stampa 1 1 2 2 3 3
- d. nessuna delle altre risposte è vera

3. (3, -0.5) Supponendo che la matrice `int a[5][3]` sia memorizzata a partire dall'indirizzo 100 e che gli oggetti di tipo `int` occupino 4 byte, l'indirizzo dell'elemento

`a[1][0]` è

- a. $100 + 1*4*3 + 0*4 = 112$
- b. $100 + 0*4*3 + 0*4 = 100$
- c. $100 + 1*4*5 + 0*4 = 120$
- d. nessuna delle altre risposte è corretta

4. (+3, -.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. Stampa `luigi marco 2000 marco`

**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: -.5**
 - **Una risposta lasciata in bianco viene calcolata: 0**
5. Contrassegnare la/le affermazione/i corretta/e:
- a) A differenza di un oggetto di tipo **int**, un oggetto di tipo **long long int** non è soggetto a problemi di *overflow*
 - b) Un oggetto di tipo **long long int** può occupare in memoria uno spazio maggiore o uguale di un oggetto di tipo **int**
 - c) Un oggetto di tipo **float** ha per definizione una precisione (numero di cifre significative dei numeri che può rappresentare) maggiore di un **long long int**
 - d) Un oggetto di tipo **double** può occupare meno memoria di un oggetto di tipo **float**
6. Assumendo che i valori di tipo **double** siano rappresentati mediante lo standard IEEE 754 in base 2:
- a) Il tipo **double** non può rappresentare tutti i numeri reali compresi nel suo intervallo di rappresentabilità
 - b) La precisione del tipo **double** non dipende dal numero di cifre riservate alla rappresentazione dell'esponente
 - c) Se **a**, **b** e **c** sono tre variabili di tipo **double**, **a** e **b** contengono due valori positivi, e si esegue l'assegnamento $c = a + b$, senza che la somma $a + b$ generi *overflow*, allora, dopo tale assegnamento, la seguente espressione logica può essere lo stesso falsa: $a == c - b$.
 - d) Nessuna delle altre risposte è vera
7. Dato un programma scritto in linguaggio C/C++
- a) Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili locali della funzione (se presenti)
 - b) Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) i parametri formali della funzione (se presenti)
 - c) Quando una funzione ne invoca un'altra, viene prima deallocato il record di attivazione della funzione chiamante e quindi creato quello della funzione chiamata
 - d) Quando inizia l'esecuzione del programma vengono allocati in memoria i record di attivazione di tutte le funzioni in esso definite

8. Dato il seguente programma:

```
main()
{
    ofstream f("nome.txt") ; // apro il file in scrittura
    f<<1<<2<<endl ;
    f<<3 ;
    ifstream f2("nome.txt") ; // riapro lo stesso file in lettura
    int i ;
    while(f2>>i) ;
        cout<<i<<" , " ;
}
```

- a) al termine del programma, il file **nome.txt** è un file binario contenente i numeri 1, 2 e 3
- b) al termine del programma, il file **nome.txt** è un file di testo contenente numeri diversi da 1, 2 e 3
- c) il programma stampa almeno **1, 2,**
- d) quando viene riaperto in lettura alla creazione dell'*ifstream* **f2**, il file **nome.txt** può non contenere tutti i caratteri che sono stati immessi nello *ofstream* **f**

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. **(6 pt)** Descrivere sintassi, semantica e motivazione degli array dinamici, in non più di 10 righe (non verrà valutata la quantità ma la qualità di quello che si scrive, ed il tentativo di scrivere in modo estremamente fitto per aumentare la quantità, così come il superamento del numero massimo di righe, comporteranno una penalità).

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 minimo.

Programmazione I
Prova scritta – 11 Settembre 2019

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 una 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 (6 pt):

Risposta alla domanda 10 (6 pt):

Programmazione I
Prova scritta – 11 Settembre 2019

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

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (6 pt):