

Programmazione I

Prova scritta - 15 settembre 2016 - 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 e sapendo che se un operatore binario ha un operando di tipo **unsigned int** e l'altro di tipo **int**, entrambi sono convertiti ad **unsigned int** prima di eseguire la relativa operazione:

```
main() {
    unsigned int i = 5 ;
    for (int j = 0 ; i - j >= 0 ; j++)
        cout<<i-j<<endl ;}
```

 - a) Esegue un ciclo infinito
 - b) Stampa i numeri da 5 a 0
 - c) Stampa i numeri da 5 a 1
 - d) Nessuna delle altre risposte è corretta
 2. (3, -0.5) Data la seguente funzione a cui viene passato un array **a** di dimensione **N**:

```
bool fun(unsigned int a[], int i, unsigned int N) {
    if ( (i < 0 || i >= N) || a[i] % 2 == 1 )
        return false ;
    return true ;
}
```

 - a) Solo se l'indice **i** è compreso tra 0 ed N-1 (estremi inclusi) la funzione controlla il valore dell'elemento **i**-esimo, e ritorna **false** se tale elemento è pari
 - b) Solo se l'indice **i** è compreso tra 0 ed N-1 (estremi inclusi) la funzione controlla il valore dell'elemento **i**-esimo, e ritorna **false** se tale elemento è dispari
 - c) Nessuna delle altre risposte è vera
 - d) La funzione può scrivere al di fuori dell'array **a**
 3. (2, -0.5) Il seguente frammento di codice **double b; b=5/2+3/2;**
 - a) causa un errore a tempo di compilazione perché il risultato dell'espressione non è rappresentabile mediante il tipo **double**
 - b) immette il valore 3.0 nella variabile **b**
 - c) causa una terminazione forzata del programma quando si prova ad eseguire l'assegnamento, perché il risultato dell'espressione non è rappresentabile mediante il tipo **double**
 - d) immette il valore 4.0 nella variabile **b**

4. (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] ;
}
```

- crea un file di testo contenente i caratteri **abc**
- stampa qualcosa di diverso da **abc**
- 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**
- 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: -.5
 - Una risposta lasciata in bianco viene calcolata: 0
5. Data la rappresentazione dei numeri in virgola mobile mediante segno, mantissa ed esponente:
- Dato un numero in virgola mobile senza parte frazionaria memorizzato tramite tale rappresentazione, si possono avere problemi di perdita di informazione nel convertirlo ad un numero di tipo **int**
 - Aumentare di una cifra la componente contenente l'esponente aumenterebbe il valore massimo memorizzabile molto di più di aumentare di una cifra la componente contenente la mantissa
 - Si possono rappresentare numeri arbitrariamente piccoli, memorizzando valori arbitrariamente bassi nell'esponente
 - Sommando un numero positivo ed uno negativo, memorizzati mediante la rappresentazione in questione, non possono mai sorgere problemi di overflow
6. Data una sequenza di elementi tutti dello stesso tipo
- Se le operazioni più frequenti sulla sequenza sono inserimenti ed estrazioni in/dal mezzo di elementi di cui si conosce la posizione in memoria, allora una lista è la struttura dati più conveniente da utilizzare, in termini di costo computazionale, per implementare la sequenza
 - Se, all'inizio della propria esecuzione, il programma è informato prima del numero effettivo di elementi della sequenza e poi del contenuto di ciascuno di tali elementi, e tale numero di elementi non varierà più durante l'esecuzione programma, allora, in termini di costo computazionale ed occupazione di memoria, la struttura dati più conveniente da utilizzare per implementare la sequenza è un array dinamico
 - Se il numero di elementi che saranno inseriti nella sequenza è noto a tempo di scrittura del programma e non avverranno mai estrazioni, allora la struttura dati più conveniente da utilizzare, in termini di semplicità di gestione, per implementare la sequenza è un array dinamico
 - Un array utilizzato per memorizzare tale sequenza può occupare più memoria di una lista che rappresenti la stessa sequenza, se alcuni degli elementi dell'array non rappresentano alcun elemento della sequenza

7. Dato il seguente programma:

```
void fun(int) ;  
int g = 10 ;  
main() { int g = 5 ; fun(g) ; cout<<g ; }
```

```
void fun(int a){ cout<<(g++)<<" " ; }
```

- a) Se eseguito, il programma stampa 10 11
- b) **Se eseguito, il programma stampa 10 5**
- c) All'interno della funzione **fun** si stampa e si modifica il valore della variabile **g** definita nella funzione **main**
- d) **All'interno della funzione **main** non è visibile 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**

8. (6 pt) In non più di 10 righe, descrivere la differenza tra il passaggio dei parametri per valore ed il passaggio dei parametri per riferimento, e menzionare i vantaggi dell'uno e dell'altro (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).

9. (7 pt) Scrivere una funzione che prende in ingresso una stringa contenente solo lettere minuscole, e ritorna una nuova stringa contenente solo i caratteri presenti almeno due volte nella stringa in ingresso. Ad esempio, presa in ingresso la stringa “*esose*”, la funzione ritornerebbe la stringa “*esse*”. Si ottiene il punteggio massimo se si realizza una soluzione con ordine di costo computazionale minimo, anche se tale soluzione richiede un'occupazione di memoria maggiore di altre.

Programmazione I

Prova scritta - 15 settembre 2016

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

Risposta alla domanda 8 (6 pt):

Risposta alla domanda 9 (7 pt):

Programmazione I

Prova scritta - 15 settembre 2016

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					3	-0,5
5						
6						
7						

Risposta alla domanda 8 (6 pt):

Risposta alla domanda 9 (7 pt):