

Programmazione I

Prova scritta - 15 gennaio 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, -.5) Tenendo presente eventuali problemi di conversione (il tipo **unsigned int** è gerarchicamente superiore al tipo **int** e qualsiasi operazione tra **unsigned int** ha per risultato un **unsigned int**) e di rappresentabilità, quale della seguenti risposte è vera riguardo al seguente frammento di codice?

```
int delta_w; unsigned int wsum;  
cin>>delta_w>>wsum;  
if (wsum - delta_w < 0) cout<<"minore"<<endl ;
```

 - a) Stampa **minore** se **delta_w** è positivo e strettamente maggiore di **wsum**
 - b) Non stampa nulla se **delta_w** è positivo e **wsum** contiene un valore strettamente minore di 0
 - c) Non stampa nulla qualsiasi sia la coppia di valori interi inserita dall'utente
 - d) Nessuna delle altre risposte è vera
2. (3, -.5) 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** 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 è vera: **a == c - b**.
 - d) Nessuna delle altre risposte è vera
3. (2, -.5) Dato il seguente programma:

```
int fun(const int a[]) { return a[0] + 1 ; }  
main() { int b[2] = {2, 3} ; cout<<fun(b) ; }
```

 - a) Quando la funzione **fun** è invocata, il contenuto dell'*array* **b** è copiato nell'*array* **a**
 - b) Quando la funzione **fun** è invocata, l'indirizzo del (primo elemento) dell'*array* **b** è copiato nel parametro formale **a**
 - c) Se la funzione **fun** contenesse istruzioni che modificano il valore di un elemento dell'*array* **a** (parametro formale) sarebbe segnalato un errore a tempo di esecuzione, in particolare l'errore sarebbe segnalato nel momento in cui si tenta di eseguire tale istruzione
 - 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: -.5**
- **Una risposta lasciata in bianco viene calcolata: 0**

4. Data una sequenza di N valori interi positivi
- a) Se, in un dato programma, la sequenza è memorizzata in un `array` con un numero di elementi pari ad N , allora non c'è modo di conoscere tale numero di elementi in un punto del programma in cui è visibile l' `array` ma non è visibile alcuna variabile in cui è memorizzato il numero di elementi dell'`array` stesso
 - b) Se la sequenza è memorizzata in una lista semplice, senza memorizzare in alcuna altra variabile il numero di elementi della sequenza, allora sono necessarie più di $O(1)$ operazioni per calcolare tale numero di elementi
 - c) Se la sequenza è memorizzata in un `array` utilizzando un terminatore per delimitare la fine della sequenza stessa, non è necessario aver memorizzato ulteriori informazioni in altre variabili per poter conoscere il numero di elementi nella sequenza
 - d) Se le operazioni più frequenti sulla sequenza sono inserimenti ed estrazioni in/dal mezzo, allora la struttura dati più conveniente da utilizzare, in termini di costo computazionale, per implementare la sequenza è un lista

5. Dato il seguente programma

```
int fun(int &g) { return 2 * (++g); }
```

```
int g = 1 ;
```

```
main() {
```

```
    char n = static_cast<int>(2.4) * fun(g) ;
```

```
    cout<<static_cast<int>(g * n) ;
```

```
}
```

- a) Se eseguito, il programma stampa **16**
- b) Nell'ultima istruzione del `main`, la conversione `static_cast<int>(g * n)` non causa perdita di informazione
- c) Nessuna delle altre risposte è vera
- d) La conversione `static_cast<int>(2.4)` causa perdita di informazione

6. 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) Se l'indice `i` è compreso tra 0 ed $N-1$ (estremi inclusi) la funzione ritorna **true**
- b) Nessuna delle altre risposte è vera
- c) Solo se l'indice `i` non 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
- d) La funzione può leggere al di fuori dell'array `a`

7. Quali delle seguenti affermazioni sono vere?

- a) Un file binario può contenere sequenze di byte uguali a quelle che possono essere contenute in un file di testo
- b) Un file di testo non contiene numeri, ma rappresentazioni grafiche di caratteri (alcuni dei quali possono eventualmente essere cifre numeriche)
- c) Nessuna delle altre risposte è vera
- d) Se si memorizza un array di caratteri in un file mediante scrittura non formattata, allora il file risultante è un file di testo contenente la sequenza dei codici dei caratteri contenuti nell'array

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. **(8 pt)** Descrivere sintassi e semantica del costrutto **for** in non più di 10 righe, utilizzando anche uno schema se lo si ritiene opportuno (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. (6 pt) Scrivere una funzione che prende in ingresso due stringhe, **a** e **b**, e cerca solo la prima occorrenza della stringa **b** all'interno della stringa **a**. Se la trova, elimina tale occorrenza dalla stringa **a**. Ad esempio, se alla funzione vengono passate le stringhe *"Piu' per meno meno"* e *"meno"*, allora la funzione trasforma la prima stringa in *"Piu' per meno"* (notare il doppio spazio tra le ultime due parole all'interno della stringa).

Programmazione I

Prova scritta - 15 gennaio 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						
5						
6						
7						

Risposta alla domanda 8 (8 pt):

Risposta alla domanda 9 (6 pt):

Programmazione I

Prova scritta - 15 gennaio 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						
5						
6						
7						

Risposta alla domanda 8 (8 pt):

Risposta alla domanda 9 (6 pt):