

Programmazione I

Prova scritta - 11 settembre 2017 - 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. (2, -0.5) Dato il seguente programma e supponendo il codice del carattere *c* sia 99, nonché facendo attenzione alle conversioni implicite per gli operatori aritmetici binari:

```
main() {  
    char a = 'd'; int b = 2 ;  
    cout<<(a - b) ;  
}
```

- a) Il programma stampa **98**
- b) Il programma stampa **b**
- c) Il programma genera un errore a tempo di compilazione perché non si può effettuare la sottrazione tra un valore di tipo *char* ed uno di tipo *int*
- d) Nessuna delle altre risposte è vera

2. (2, -0.5) Dato il seguente programma:

```
main()  
{  
    int a ; cin>>a ; bool b = a > 0 ;  
    if ( (a <= 0) == (! b) )  
        cout<<a ;  
}
```

- a) Se l'utente immette **0** su *stdin* il programma non stampa nulla
- b) Il programma contiene un errore di gestione della memoria
- c) Nessuna delle altre risposte è vera
- d) **Qualsiasi valore l'utente immetta su *stdin*, il programma stampa il valore immesso**

3. (2, -0.5) Dato il seguente programma:

```
int globale = 2;  
void fun(int &rif_locale)  
{ rif_locale++ ; cout<<rif_locale<<" "<<globale ; }  
main()  
{ globale = 3; fun(globale) ;}
```

- a) Se eseguito, il programma stampa **4 3**
- b) **Se eseguito, il programma stampa 4 4**
- c) Nessuna delle altre risposte è vera
- d) Se eseguito, il programma stampa **4 2**

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. Facendo attenzione ad eventuali problemi di *overflow* e supponendo che la costante carattere 'w' contenga un valore maggiore di zero, il seguente frammento di codice:
- ```
int i; cin>>i;
if (static_cast<unsigned char>(i) > 'w')
 cout<<"Maggiore" ;
```
- a) stampa o meno **Maggiore** in base al valore assunto dalla variabile **i**
- b) può non stampare **Maggiore** per qualche valore di **i** maggiore del valore della costante carattere 'w'
- c) può stampare **Maggiore** per qualche valore di **i** minore di zero
- d) nessuna delle altre risposte è corretta
5. Data una sequenza di N elementi tutti dello stesso tipo
- a) Se la sequenza è memorizzata in *array* e si vuole scrivere il valore di un elemento di cui si conosce solo la posizione nella sequenza (si sa che è l'elemento i-esimo, con  $i < N$ ), allora la lettura può essere effettuata a costo  $O(1)$
- b) Se la sequenza è memorizzata in una lista semplice e si vuole leggere il valore di un elemento di cui si conosce solo la posizione nella sequenza (si sa che è l'elemento i-esimo, con  $i < N$ ), allora la lettura può essere effettuata a costo  $O(1)$
- c) Se la sequenza è memorizzata in un *array* utilizzando un terminatore per delimitare la fine della sequenza stessa, non è necessario mantenere ulteriori informazioni per poter accedere sequenzialmente a tutti gli elementi della sequenza
- d) A seconda della dimensione dei singoli elementi (uguale per tutti gli elementi) della sequenza e del valore di N, memorizzare la sequenza in una lista potrebbe essere più efficiente, in termini di occupazione della memoria, di memorizzare la sequenza in un *array* con terminatore (ossia un array in cui si utilizza un terminatore per delimitare la fine della sequenza)
6. Assumendo che i numeri reali siano rappresentati mediante lo standard IEEE 754 in base 2:
- a) Dati due numeri reali, entrambi rappresentati in modo esatto, e tali che il primo numero è il doppio dell'altro, si ha che le mantisse dei due numeri sono uguali
- b) Dati due numeri reali rappresentabili in modo esatto e che differiscono solo per il segno, le loro rappresentazioni hanno la stessa mantissa
- c) Nessuna delle altre risposte è vera
- d) Se due numeri reali sono rappresentabili in modo esatto, e la loro divisione ha un valore non inferiore al numero reale minimo rappresentabile, allora anche tale divisione è rappresentabile in modo esatto
7. Dato il seguente programma:
- ```
const int N = 1000 ; struct ss { int b; char a[N] ; } ;
void fun(ss d) { d.a[N/2] = 15 ; d.b = 4 ; cout<<d.b ; }
main()
{ ss c ; c.b = 10 ; c.a[N/2] = 2 ;
  fun(c) ; cout<<" "<<c.b<<" "<<static_cast<int>(c.a[N/2]) ;
}
```

- a) Quando **fun** è invocata il contenuto del parametro attuale **c** è copiato nel parametro formale **d**
- b) In quanto al campo **a** del parametro attuale **c** passato alla funzione **fun**, l'array in esso contenuto è copiato nel campo **a** del parametro formale **d**.
- c) Un passaggio per riferimento sarebbe stato più conveniente in termini di tempo di esecuzione
- d) Il programma stampa: **4 10 2**

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. (7 pt) Descrivere la differenza tra output formattato ed output non formattato in non più di 10 righe, e rispondere alla seguente domanda “E’ possibile scrivere un file di testo utilizzando output non formattato, e se sì come?” (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 prenda in ingresso una stringa, che si assume contenere solo lettere minuscole, e ritorni vero se la stringa contiene caratteri tutti diversi tra loro, falso altrimenti. Ad esempio, la funzione ritorna vero per la stringa *“pluto”* e falso per la stringa *“pippo”*. Si ottiene il punteggio massimo se si realizza una funzione che scorre sia la stringa che altri eventuali array di appoggio al più una volta.

Programmazione I

Prova scritta - 11 settembre 2017

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

Risposta alla domanda 8 (7 pt):

Risposta alla domanda 9 (7 pt):

Programmazione I

Prova scritta - 11 settembre 2017

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

Risposta alla domanda 8 (7 pt):

Risposta alla domanda 9 (7 pt):