

Programmazione I

Prova scritta - 9 gennaio 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. (3, -.5) Tenendo presente eventuali problemi di conversione implicita e di rappresentabilità, il seguente frammento di codice

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

- a) Può stampare maggiore anche se **delta_w** è positivo
- b) Stampa maggiore solo se **delta_w** è negativo e l'espressione algebrica **-delta_w > wsum** è vera
- c) Stampa maggiore se **delta_w** è negativo e **wsum** contiene un valore negativo
- d) Nessuna delle altre risposte è vera

2. (2, -.5) Il seguente programma:

```
int a = 10, b = 20 ;  
int &fun(bool f)  
{ if (f) return a ;  
  else return b ; }  
main()  
{ int &c = fun(true) ; a++ ; b-- ;  
  cout<<c; }
```

- a) Stampa 10
- b) Stampa 11
- c) Stampa 19
- d) Nessuna delle altre risposte è corretta

3. (2, -0,5) La memoria occupata da un oggetto **X** allocato dinamicamente all'interno di una funzione

- a) è allocata all'inizio dell'esecuzione della funzione;
- b) è liberata se si assegna l'indirizzo di un nuovo oggetto dinamico al puntatore che contiene l'indirizzo di **X** in memoria;
- c) è liberata se viene deallocato il puntatore che contiene l'indirizzo di **X** in memoria;
- d) se non esplicitamente deallocata, è liberata solo al termine del programma.

4. (3, -0.5) Dato il seguente programma

```
main()
{
  int i = 10 ; char s[10] ; ifstream f("dati.txt");
  f>>i ; f>>s ;
  cout<<s<<" "<<i<<endl ;
}
```

e supponendo che il file *dati.txt* esista e contenga la sequenza di caratteri **7 sette** (c'è uno spazio tra il carattere **7** e la stringa *sette*) seguiti dal carattere *newline*:

- Il programma memorizza il valore **7** nella variabile *i*, poi si blocca indefinitamente sull'istruzione **f>>s** ; perché manca il carattere *newline* tra il carattere **7** e la stringa **sette** all'interno del file *dati.txt*
- Nessuna delle altre risposte è corretta
- Il programma stampa **sette 7**
- L'*ifstream f* va in stato di errore sull'istruzione **f>>s** ;

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 una sequenza di *N* valori interi positivi

- Aggiungere un elemento in testa alla sequenza comporta un numero di passi proporzionale ad *N*, se la sequenza è implementata mediante una lista semplice
- Accedere all'*i*-esimo elemento della sequenza comporta un numero di passi costante, indipendentemente dal valore di *N*, se la sequenza è implementata mediante una lista semplice
- Si può memorizzare la sequenza in un *array* utilizzando un terminatore per delimitare la fine della sequenza stessa, ed in tal caso non è necessario mantenere ulteriori informazioni per poter accedere sequenzialmente a tutti gli elementi della sequenza
- La struttura dati più efficiente, in termini di occupazione di memoria, in cui si potrebbero memorizzare tali valori è un array di *N* elementi

6. Assumendo che i valori di tipo **double** siano rappresentati, mediante lo standard IEEE 754 in base 2, con una precisione di 53 cifre in base 2:

- Un qualsiasi numero reale che sia strettamente maggiore di zero e minore di 1, e che, nella sua rappresentazione in base 2, abbia un numero di cifre dopo la virgola inferiore a 53, è rappresentabile in modo esatto mediante il tipo **double**.
- Un qualsiasi numero reale che sia strettamente maggiore di zero e minore di 1, e che, nella sua rappresentazione in base 10, abbia una sola cifra dopo la virgola, è rappresentabile in modo esatto mediante il tipo **double**.
- Mediante il tipo di dato **double** non si possono rappresentare in modo esatto tutti i numeri reali compresi nell'intervallo di rappresentabilità del tipo di dato stesso.
- Esistono numeri strettamente maggiori di zero e minori di 1, e con più di 53 cifre dopo la virgola nella loro rappresentazione in base 2, che si possono rappresentare in modo esatto mediante il tipo **double**

7. Dato il seguente programma

```
int g = 1 ;

double fun(int n)
{ g = g + 2 ; return n * 5 ; }
```

```

main()
{
    const double g = 2 ;
    double l = static_cast<int>(fun(g / 2)) / 2 ;
    cout<<static_cast<int>(l * g) ;
}

```

- a) Se eseguito, il programma stampa **4**
- b) La conversione esplicita presente nell'espressione a destra dell'inizializzazione nella penultima istruzione della funzione *main* causa perdita di informazione
- c) Se non fosse stata presente la conversione esplicita nell'espressione a destra dell'inizializzazione nella penultima istruzione della funzione *main* il programma avrebbe stampato un valore diverso
- d) Il programma causa un errore a tempo di compilazione, perché la prima istruzione della funzione *fun* prova a modificare la costante *g* definita nella prima istruzione della funzione *main*

8. Dato un ciclo **for** o **while**,

- a) mediante l'istruzione **break** si esce dal ciclo più interno contenente l'istruzione e da altri eventuali cicli esterni;
- b) mediante l'istruzione **break** si esce solo dal ciclo più interno contenente l'istruzione;
- c) mediante l'istruzione **continue** si termina l'esecuzione del ciclo più interno contenente l'istruzione e si continua l'esecuzione degli eventuali cicli più esterni;
- d) mediante l'istruzione **continue** non si termina l'esecuzione del ciclo che contiene l'istruzione.

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) 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).

10. (7 pt) Scrivere una funzione che prende in ingresso un vettore di interi e lo modifica trasformandolo in un vettore che contiene gli stessi valori interi ordinati in senso crescente, tranne i valori che compaiono più di una volta nel vettore originale. Ad esempio, se la funzione prende in ingresso il vettore [4, 6, 3, 4, 1, 10, 6, 6], allora trasforma il vettore in [1, 3, 10]. Il punteggio ottenuto è proporzionale all'efficienza della funzione implementata, in termini sia di numero di passi che di occupazione di memoria.

Programmazione I

Prova scritta - 9 gennaio 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					3	-0,5
2					2	-0,5
3					3	-0,5
4					2	-0,5
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):

Programmazione I

Prova scritta - 9 gennaio 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					3	-0,5
2					2	-0,5
3					2	-0,5
4					3	-0,5
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):