

Programmazione I

Prova scritta - 14 giugno 2010 - 1h30min

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 calcolata: 0

1. (2, -0.5) Dato il seguente programma:

```
void fun(int a[], int n)
{ for (int i = 0 ; i < n ; i++) cin>>a[i] ;}
```

```
main() { int c[3] ; fun(c, 3) ; }
```

- a) La funzione **fun** modifica solo una copia locale dell'array che le viene passato
 - b) La funzione **fun** modifica l'array **c** che le viene passato
 - c) La variabile **c** ha classe di memorizzazione statica
 - d) Nessuna delle altre risposte è corretta
2. (2, -0.5) Contrassegnare l'affermazione corretta riguardo gli array:
- a) Un array può immagazzinare valori di tipo diverso
 - b) L'indice di un array può essere del tipo di dato **double**
 - c) È un errore inserire in una lista di inizializzazione di un array più elementi di quanti ne possa contenere l'array stesso
 - d) È un errore inserire in una lista di inizializzazione di un array meno elementi di quanti ne possa contenere l'array stesso
3. (2, -0.5) Il seguente frammento di codice `int 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 **int**
 - b) immette il valore 3 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 **int**
 - d) immette il valore 4 nella variabile **b**
4. (3, -0.5) Il seguente frammento di codice:
- ```
for (int i=0; i <11 ; i++) { if (i==7) i--; cout<<i*2<<" " ;}
```
- a) Stampa i numeri pari fra 0 e 12 ed esce dal ciclo quando **i** è uguale a 7
  - b) Stampa i numeri dispari fra 0 e 20
  - c) Nessuna delle altre risposte è vera
  - d) Stampa i numeri pari fra 0 e 20, stampando per due volte di seguito il numero 14

**PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE -  
Ogni domanda può avere una o più risposte CORRETTE.**

- Ogni risposta esatta viene calcolata: +1
  - Ogni risposta errata viene calcolata: -0.5
  - Una risposta lasciata in bianco viene calcolata: 0
5. Due algoritmi equivalenti:
- a) Forniscono lo stesso risultato per qualsiasi insieme di dati in ingresso
  - b) Hanno lo stesso codice
  - c) Prevedono gli stessi passi
  - d) A parità di dati in ingresso possono avere tempi di esecuzione diversi
6. Dato il seguente programma:
- ```
1:float b = 3.5;
2:float fun(float &a)
3:{
4:    int i ;
5:    for (i = 0 ; i < 2 ; i++)
6:        a *= 2 ;
7:    return a + b + i ;
8:}
9:
10:main()
11:{
12:    float b = 1.5 ;
13:    float c = fun(b) ;
14:    cout<<static_cast<int>(c + b)<<endl ;
15:}
```
- a) la funzione **fun** modifica il valore della variabile **b** definita alla riga 12
 - b) il parametro formale **a** definito alla riga 2 non è visibile alla riga 14
 - c) la variabile **i** definita alla riga 4 non è visibile alla riga 7
 - d) il programma stampa **17**
7. Per definire una costante PIGRECO di valore 3.1415:
- a) Si può utilizzare l'istruzione `const float PIGRECO = 3.1415;`
 - b) Si può utilizzare l'istruzione `const int PIGRECO = 3.1415;`
 - c) Si può utilizzare l'istruzione `float PIGRECO = 3.1415;`
 - d) Si può utilizzare la sequenza di istruzioni `const float PIGRECO; PIGRECO = 3.1415;`
8. Indicare quale/i delle seguenti affermazioni sulle stringhe è/sono corrette;
- a) il terminatore di stringa non è indispensabile nel caso di un'allocazione dinamica della stringa
 - b) il carattere '\n' non può essere utilizzato per indicare il fine stringa
 - c) le stringhe statiche non vengono memorizzate nello heap
 - d) in funzione del numero di caratteri contenuti in una stringa, lo spazio da essa occupato in memoria può essere minore delle dimensioni dell'array di caratteri utilizzato per rappresentarla

9. 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<<" , " ;
}
```

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

10. Dato un programma scritto in linguaggio C/C++

- L'esecuzione di una istruzione di allocazione dinamica della memoria non provoca nessuna variazione delle dimensioni del record di attivazione della funzione contenente tale istruzione
- Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni delle variabili locali inizializzate
- Per deallocare un record di attivazione è necessario reinizializzare il contenuto delle celle di memoria precedentemente occupate dal record stesso
- Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili globali a cui accede la funzione

11. Dato il seguente programma

```
bool fun(int &a)
{
    if (a > 10) { ++a ; return true ; }
    return false ;
}

main()
{
    int m = 22 ;

    if (m < 30 && fun(m)) cout<<(m*2) ;
    else cout<<m ;
}
```

- Se eseguito, il programma stampa 46
- Se eseguito, il programma stampa 22
- Dopo la sua inizializzazione, il valore della variabile **m** non è mai modificato durante l'esecuzione del programma
- Il valore della variabile **m** è modificato durante l'esecuzione della funzione **fun** all'interno del **main**

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

12. (4 pt.) Cosa scrive su *stdout* il seguente programma?

```
int funz(int a, int b);
int r = 0;

main()
{
    int n1=10, n2=20, ris=0;
    cout<<n1<<"<<n2<<endl;
    ris = funz(n1, n2);
    ris += r;
    cout<<n1<<"<<n2<<"<<ris<<endl;
}

int funz(int a, int b)
{
    r = (a + b)%2;
    if (r == 0)
        r = (a + b)/2;
    cout<<r<<endl ;
    return r+1;
}
```

13. **(5 pt)** Scrivere una funzione che prenda in ingresso una stringa, e crei e ritorni una nuova stringa, con lo stesso contenuto dell'originale, ruotato però di una posizione verso destra. Ad esempio, se la stringa di ingresso fosse *"abcd"*, verrebbe creata e ritornata la stringa *"dabc"*.

Programmazione I

Prova scritta - 14 giugno 2010

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

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 un 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					3	-0,5
5						
6						
7						
8						
9						
10						
11						

Risposta alla domanda 12 (4 pt):

Risposta alla domanda 13 (5 pt):

Programmazione I

Prova scritta - 14 giugno 2010

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

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 un 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					3	-0,5
5						
6						
7						
8						
9						
10						
11						

Risposta alla domanda 12 (4 pt):

Risposta alla domanda 13 (5 pt):