

# Programmazione I

## Prova scritta - 11 febbraio 2011 - 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; cin>>n ; for (int i = 0 ; i < n ; i++) cin>>a[i] ;}

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

- a) La funzione **fun** modifica solo una copia locale dell'array **c** che le viene passato
  - b) La funzione **fun** modifica solo una copia locale dell'array **c** che le viene passato
  - c) La variabile **n** ha classe di memorizzazione automatica
  - d) Nessuna delle altre risposte è corretta
2. (2, -0.5) Contrassegnare l'affermazione corretta riguardo gli array:
- a) E' impossibile definire un array contenente oggetti di tipo eterogeneo
  - b) I valori degli indici degli elementi partono da 1
  - c) Se un array non viene inizializzato, ai suoi elementi vengono assegnati valori casuali
  - d) Nessuna delle altre risposte è corretta
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 frammento di codice:
- ```
for (int i=5; i >0 ; i--) { if (i==3) i++; cout<<i*2<<" " ;}
```
- a) Stampa 10 8 6
  - b) Contiene un ciclo infinito
  - c) Nessuna delle altre risposte è vera
  - d) Stampa 10 8 8 6 4 2

### 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 a parità 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 ;
13:    float c = fun(b) ;
14:    cout<<static_cast<int>(c)<<endl ;
15:}

```

- a) la funzione **fun** non 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 **9**
7. Per definire una costante PIGRECO di valore 3.1415:
- a) Si può utilizzare l'istruzione `const double PIGRECO = 3.1415;`
  - b) Si può utilizzare l'istruzione `const int PIGRECO = 3.1415;`
  - c) Si può utilizzare l'istruzione `double PIGRECO = 3.1415;`
  - d) Si può utilizzare la sequenza di istruzioni `double PIGRECO; PIGRECO = 3.1415;`
8. Indicare quale/i delle seguenti affermazioni sulle stringhe è/sono corrette;
- a) il terminatore di stringa non è indispensabile per le stringhe dinamiche
  - b) il carattere '\n' non può essere utilizzato per indicare il fine stringa
  - c) le stringhe rappresentate mediante array dinamici 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 è indipendente dal numero e le dimensioni delle variabili locali inizializzate
- Un record di attivazione relativo ad una data funzione **fun** contiene anche le informazioni necessarie per far riprendere l'esecuzione della funzione chiamante a partire dall'istruzione successiva alla chiamata di **fun** stessa
- Il record di attivazione relativo ad una data funzione non contiene 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
- Se eseguito, il programma stampa 44
- 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 + 1)%2;
    if (r == 1)
        r = (a + b)/2;
    cout<<r<<endl ;
    return r+1;
}
```

13. (4 pt) Scrivere una funzione che prenda in ingresso una stringa, e ritorni una nuova stringa, con lo stesso contenuto dell'originale, ruotato però di una posizione verso sinistra. Ad esempio, se la stringa di ingresso fosse "abcd", verrebbe creata e ritornata la stringa "bcda".



# Programmazione I

## Prova scritta - 1 febbraio 2011

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/<br>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 (4 pt):**



# Programmazione I

## Prova scritta - 1 febbraio 2011

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/<br>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 (4 pt):**