

# Programmazione I

## Prova scritta - 25 settembre 2014 - 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) Date le istruzioni `cout<<'4'<<'1'<<endl ;` e `cout<<41<<endl ;` e supponendo che l'operatore di uscita sia configurato per stampare i numeri interi in base 10.
    - a) Tali istruzioni immettono sullo *stdout* la stessa sequenza di codici carattere dell'istruzione `cout<<4<<1<<endl ;`
    - b) Nessuna delle altre risposte è vera
    - c) La seconda istruzione immette su *stdout* un numero minore di byte rispetto alla prima
    - d) La seconda istruzione immette su *stdout* un numero maggiore di byte rispetto alla prima
  2. (3, -0.5) Dato il seguente programma:

```
main() {
    int i ; cin>>i ; unsigned int u = 25 ;
    cout<<u+static_cast<unsigned int>(i) ;
}
```

    - a) per qualsiasi numero intero, positivo o negativo, immesso dall'utente, il programma stampa un numero non negativo
    - b) per qualsiasi numero intero *i*, positivo o negativo, immesso dall'utente, il programma stampa il valore dell'espressione  $25 - i$
    - c) se l'utente immette **-25** da *stdin*, il programma viene terminato forzatamente
    - d) nessuna delle altre risposte è vera
  3. (3, -0.5) Dato il seguente programma:

```
main()
{
    int a ; cin>>a ; bool b = a > 5 ;
    if ( ( a > 0 ) == ( ! b ) ) cout<<a ;
}
```

    - a) Se l'utente immette **5** su *stdin* il programma stampa **5**
    - 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 non stampa nulla

4. (2, -0.5) Dato il seguente programma

```
main() {
    int i = 0 ; cin>>i ;
    if (i == '1') cout<<"Primo" ;
    else cout<<"Secondo" ;
}
```

- a) Se l'utente immette da *stdin* un valore diverso dal codice numerico usato per rappresentare il carattere 1, il programma stampa **Secondo**
- b) Se l'utente immette 1 da *stdin* il programma stampa **Primo**
- c) Se l'utente immette z da *stdin* il programma entra in un ciclo infinito
- 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: -0.5
- Una risposta lasciata in bianco viene calcolata: 0

5. Dato il seguente programma:

```
int *a, num_elem ;
void inserisci(int n)
{ if (a == 0) a = new int[10] ; a[num_elem] = n ; num_elem++ ;}
void stampa() { for (int i = 0 ; i < num_elem ; i++) cout<<a[i]<<" " ; }
main(){ inserisci(2) ; stampa() ; inserisci(4) ; stampa() ; }
```

- a) il programma stampa 2 2 4
- b) il programma può stampare qualcosa di diverso da 2 2 4
- c) il programma non contiene errori di *memory leak*
- d) il programma non contiene errori di gestione della memoria diversi dal *memory leak*

6. Durante l'esecuzione del programma in linguaggio macchina ottenuto dalla compilazione di un programma scritto in linguaggio C/C++

- a) In ogni istante le dimensioni dello *stack* sono proporzionali al numero di funzioni invocate e non ancora terminate
- b) La scrittura al di fuori di un *array* locale ad una funzione non comporta l'aumento delle dimensioni del record di attivazione della funzione
- c) Il valore iniziale di una variabile non inizializzata e locale ad una funzione diversa dal *main* dipende solo dai valori memorizzati nella sequenza di celle di memoria riservate al programma al suo avvio
- d) Se in un dato istante non vi è più memoria sufficiente per creare un nuovo record di attivazione, allora viene liberata memoria dinamica per fargli posto

7. Data la rappresentazione dei numeri in virgola mobile mediante segno, mantissa ed esponente:

- a) Dato un numero senza parte frazionaria memorizzato tramite tale rappresentazione, si possono avere problemi di perdita di informazione nel convertirlo ad un numero di tipo **int**
- b) Aumentare di una cifra la componente contenente l'esponente aumenterebbe il valore massimo memorizzabile molto di più di aumentare di una cifra la componente contenente la mantissa
- c) Memorizzando valori arbitrariamente piccoli nell'esponente, si possono rappresentare numeri arbitrariamente piccoli
- d) Sommando un numero positivo ed uno negativo, memorizzati mediante la rappresentazione in questione, non possono mai sorgere problemi di *overflow*

8. Dato il seguente programma, e supponendo che non sorga alcun problema di approssimazione dovuto alla precisione limitata del tipo **double**:

```
1: double a = 5.2 ;  
2:  
3: void fun(int a, int b)  
4: { a -= b ; }  
5: main()  
6: { fun(a, 4) ; cout<<a<<endl ; }
```

- a) il programma stampa **5.2**
- b) la variabile **a** definita alla riga 1 non ha tempo di vita pari all'intero programma;
- c) prima del decremento, la variabile **a** riferita alla riga 4 ha un valore diverso da 5.2;
- d) la variabile **a** definita alla riga 1 è visibile alla riga 4

### 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)** Descrivere sintassi, semantica e vantaggi del tipo enumerato, in non più di otto righe più eventuali frammenti di codice o righe scritte in una qualche notazione (a supporto della descrizione della sintassi).

10. (7 pt) Scrivere una funzione che prende in ingresso una matrice di interi con un numero di colonne definito a tempo di scrittura del programma. Ogni riga della matrice contiene un vettore di valori positivi, terminato dal valore -1. La funzione scambia il contenuto della riga della matrice contenente il vettore più lungo con quello della riga contenente il vettore più corto (la funzione quindi modifica la matrice). Ad esempio, se la matrice in ingresso è  $\begin{bmatrix} 2 & 5 & 1 & 3 \\ 7 & 9 & 6 & 2 \end{bmatrix}$ , la funzione la modifica in  $\begin{bmatrix} 2 & 5 & 1 & 3 \\ 7 & 9 & 6 & 2 \end{bmatrix}$ . Nel caso in cui più righe contengano vettori di pari lunghezza minima e/o più righe contengano vettori di pari lunghezza massima, qualsiasi coppia scelta per lo scambio va bene. Si ottiene il punteggio massimo se la funzione non utilizza un *array* di appoggio per implementare lo scambio.



# Programmazione I

## Prova scritta - 25 settembre 2014

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 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 |          |   |   |   | 3                  | -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 - 25 settembre 2014

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/<br>Penalità |      |
|---|----------|---|---|---|--------------------|------|
|   | A        | B | C | D |                    |      |
| 1 |          |   |   |   | 2                  | -0,5 |
| 2 |          |   |   |   | 3                  | -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):**