

# Programmazione I

## Prova scritta - 30 giugno 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) Quale delle seguenti affermazioni è vera?
    - a) In C non esiste il passaggio di un oggetto per riferimento, ma si può emularlo passando l'indirizzo dell'oggetto stesso
    - b) In C non esiste il passaggio per riferimento e non c'è nessun modo per emularlo
    - c) Dato un parametro formale di tipo puntatore ad un certo tipo **T**, allora, all'atto del passaggio dell'indirizzo di un oggetto di tipo **T** come parametro attuale nella posizione corrispondente a tale parametro formale, non avviene la copia di tale indirizzo nel parametro formale
    - d) Nessuna delle altre affermazioni è vera
  2. (3, -.5) Assumendo: 1) che nel seguente programma l'operazione di apertura del file in lettura nella funzione **leggi** abbia successo, e 2) che nel file sia correttamente memorizzato l'*array* scritto dalla funzione **scrivi**

```
void scrivi(const char b[], int m){ ofstream f("numeri");
    for (int i = 0 ; i < m ; i++)
        if (!(f<<b[i])) break ; }
void leggi(int c[], int n){ ifstream f("numeri");
    for (int i = 0 ; i < n ; i++)
        if (!(f>>c[i])) break ; }
main() {
    int c[] = {2, 3}; char b[] = {'6', ' ', '2'};
    scrivi(b, 3) ;
    leggi(c, 2) ;
    for (int i = 0 ; i < 2 ; i++) cout<<c[i]<<" " ;
}
```

    - a) il programma contiene un errore di gestione della memoria
    - b) il programma stampa 2 3
    - c) il programma stampa 6 2
    - d) nessuna delle altre risposte è corretta

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

```
int *a, num_elem ;
void distruggi(int *b) { delete [] b; num_elem = 0 ; }
void inserisci(int n) { a[num_elem] = n ; num_elem++ ; }
void stampa() { for (int i = 0 ; i < num_elem ; i++) cout<<a[i]<<" " ;
}

main() {
    a = new int[10] ;
    inserisci(1) ; stampa() ;
    int *b = a ;
    distruggi(b) ;
    inserisci(4) ; stampa() ; cout<<b[0]<<endl ;
}
```

- a) il programma stampa 1 4 4  
b) nessuna delle altre risposte è vera  
c) il programma contiene un errore di *memory leak*  
d) il programma contiene un errore di gestione della memoria diverso dal *memory leak*
4. (2, -0.5) Il seguente programma:
- ```
void fun(int &c) { cout<<c; c = 3 ; }
main() { int a = 2 ; int b = a ; fun(a) ; cout<<b ; }
```
- a) contiene un errore di gestione della memoria  
b) stampa 22  
c) stampa 23  
d) nessuna delle altre affermazioni è vera
5. (3, -0.5) Dato il seguente programma e sapendo che se un operatore binario ha un operando di tipo **unsigned int** e l'altro di tipo **int**, entrambi sono convertiti ad **unsigned int** prima di eseguire la relativa operazione:
- ```
main() {
    unsigned int i = 5 ;
    for (int j = 0 ; i - j >= 0 ; j++)
        cout<<i-j<<endl ;}
```
- a) Esegue un ciclo infinito  
b) Stampa i numeri da 5 a 0  
c) Stampa i numeri da 5 a 1  
d) Nessuna delle altre risposte è corretta

**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
6. Dato un programma scritto in linguaggio C/C++
- a) Il tempo necessario per inizializzare il contenuto di un record di attivazione è indipendente dal numero e le dimensioni delle variabili locali inizializzate
  - b) Il record di attivazione relativo ad una data funzione non contiene le variabili globali a cui accede la funzione
  - 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
  - d) 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

7. Nel seguente programma:

```
1:int a = 1;
2:
3:fun()
4:{
5:    int i ;
6:    for (i = 0 ; i < 5 ; i++) {
7:        double a = i / 2.0 ;
8:        cout<<a * i ;
9:    }
10:    cout<<a + i ;
11;}
12:
13:main()
14:{
15:    double b = 1.4 ;
16:    fun() ;
17:    cout<<a + b<<endl ;
18:}
```

- a) la variabile **a** definita alla riga 1 ha tempo di vita pari all'intero programma;
- b) la variabile **b** definita alla riga 15 non ha visibilità pari all'intero programma;
- c) la variabile **a** definita alla riga 1 non è visibile alla riga 8;
- d) la variabile **b** definita alla riga 15 ha tempo di vita pari all'intero programma.

### 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. (6 pt) Descrivere l'idea alla base e lo schema generale della rappresentazione in virgola mobile dei numeri reali, in non più di otto righe più eventuali schemi.

9. (6 pt) Scrivere una funzione che prende in ingresso un array A di caratteri, di lunghezza L almeno pari a due, ed esegue iterativamente i seguenti passi: legge una parola da *stdin*, e, se non più lunga di (L-2) caratteri, la inserisce nell'array A con un '\0' in fondo. Se invece la parola è più lunga di (L-2) caratteri, allora la funzione inserisce due '\0' in fila nell'array e termina. Se non è terminata nell'iterazione precedente, la funzione legge una seconda parola da *stdin*, e, se non più lunga del numero di caratteri dell'array A non ancora utilizzati meno uno, la inserisce nell'array A con '\0' in fondo, a partire dall'elemento subito successivo a quello in cui ha inserito il precedente '\0'. Se invece la parola è più lunga del numero di caratteri rimanenti meno uno, allora la funzione inserisce solo un '\0' e termina. Alla successiva iterazione la funzione fa di nuovo la stessa cosa, e così via. Ad esempio, se L=15 e si inseriscono le tre parole *gatto*, *topo*, *prende*, allora l'array conterrebbe i seguenti caratteri fino al doppio terminatore:

'g', 'a', 't', 't', 'o', '\0', 't', 'o', 'p', 'o', '\0', '\0'.



# Programmazione I

## Prova scritta - 30 giugno 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/ Penalità	
	A	B	C	D		
1					2	-0,5
2					3	-0,5
3					4	-0,5
4					2	-0,5
5					3	-0,5
6						
7						

**Risposta alla domanda 8 (6 pt):**

**Risposta alla domanda 9 (6 pt):**



# Programmazione I

## Prova scritta - 30 giugno 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/ Penalità	
	A	B	C	D		
1					2	-0,5
2					3	-0,5
3					4	-0,5
4					2	-0,5
5					3	-0,5
6						
7						

**Risposta alla domanda 8 (6 pt):**

**Risposta alla domanda 9 (6 pt):**