

# Programmazione I

## Prova scritta - 1 settembre 2015 - 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. (3, -0.5) Dato il seguente programma

```
int g = 1 ;
```

```
int fun() { ++g ; return 7 ; }
```

```
main()
```

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

- a) Se eseguito, il programma stampa 7
- b) Se eseguito, il programma stampa 6
- c) Nessuna delle altre risposte è corretta
- d) Nell'ultima istruzione del **main**, la conversione esplicita causa perdita di informazione

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

```
void fun(int a) { cout<<a ; }  
main() { char car = 'a' ; fun(car) ; }
```

- a) Se eseguito, il programma stampa il carattere **a** su *stdout*
- b) Il programma contiene un errore di gestione della memoria
- c) Nessuna delle altre risposte è corretta
- d) Il programma termina prima di stampare qualcosa su *stdout*

4. (3, -.5) Il seguente programma:

```
main()
{
  int vett[] = {1, 3, 1, 5} ;
  int *p = vett ;
  for (int i = 0 ; i < 4 ; i++)
    p[i] = vett[i] - p[i] ;
  for (int i = 0 ; i < 4 ; i++)
    cout<<vett[i]<<" " ;
}
```

- a) stampa 1 3 1 5
- b) stampa 0 0 0 0
- c) contiene un errore di gestione della memoria
- 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. Data una sequenza di  $N$  valori interi positivi

- a) Aggiungere un elemento in testa alla sequenza ha costo  $O(N)$  se la sequenza è implementata mediante una lista semplice
- b) Accedere all'  $i$ -esimo elemento della sequenza ha costo  $O(1)$  se la sequenza è implementata mediante una lista semplice
- c) Se la sequenza è memorizzata in un *array* utilizzando un terminatore per delimitare la fine della sequenza stessa, non è necessario mantenere ulteriori informazioni per poter accedere sequenzialmente a tutti gli elementi della sequenza
- d) 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 seguenti numeri siano rappresentati mediante lo standard IEEE 754 in base 2:

- a) Dati due numeri, entrambi rappresentati in modo esatto, e tali che il primo numero è il doppio dell'altro, si ha che le mantisse dei due numeri sono uguali
- b) Dati due numeri rappresentabili in modo esatto e che differiscono solo per il segno, le loro rappresentazioni hanno la stessa mantissa
- c) Nessuna delle altre risposte è vera
- d) Se due numeri sono rappresentabili in modo esatto, ed il loro prodotto non genera *overflow*, allora anche tale prodotto è rappresentabile in modo esatto

7. Dato il seguente programma:

```
int g = 10 ;

bool fun() { ++g ; return true ;}

main() {
  const int m = 22 ;

  if ((m % 2 == 0) && fun()) cout<<-g ;
  else cout<<g ;
}
```

- a) Se eseguito, il programma stampa **10**
- b) Se eseguito, il programma stampa **-11**
- c) Il valore della variabile globale **g** viene modificato durante l'esecuzione del programma (dopo la sua inizializzazione)
- d) Durante l'esecuzione del programma, è invocata anche la funzione **fun**

8. Nel seguente frammento di codice:

```
1: funzione1()
2: {
3:   int bz=5;
4:   float ak=3.2;
5:   int ps=9;
6:   {
7:     int ps;
8:     int cj;
9:     ps=bz+5;
10:  }
11: }
```

- a) le variabili **bz**, **ak**, **ps** definite nelle righe 3-5 hanno tempo di vita e scope relativo a tutto il blocco della **funzione1()**
- b) le variabili **bz**, **ak**, **ps** definite nelle righe 3-5 hanno tempo di vita relativo a tutto il blocco della **funzione1()**
- c) le variabili **bz**, **ak**, **cj** hanno scope relativo a tutto il blocco della **funzione1()** eccetto il blocco interno
- d) le variabili **bz**, **ak** hanno scope relativo a tutto il blocco della **funzione1()**, mentre la variabile **ps** definita riga 5 ha scope relativo a tutto il il blocco della **funzione1()** eccetto il blocco interno

### **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 output formattato ed output non formattato (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. (4 pt) Senza utilizzare alcuna funzione di libreria per le stringhe, scrivere una funzione che prenda un ingresso una stringa e, all'interno della stringa, trasformi ciascuna lettera maiuscola in minuscola, e viceversa (lasciando inalterati tutti i caratteri che non sono lettere). Ad esempio, la funzione trasformerebbe la stringa *"abGhdJJ3"* in *"ABgHDjj3"*.



# Programmazione I

## Prova scritta - 1 settembre 2015

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					3	-0,5
3					3	-0,5
4					3	-0,5
5						
6						
7						
8						

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

**Risposta alla domanda 10 (4 pt):**



# Programmazione I

## Prova scritta - 24 giugno 2015

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					3	-0,5
3					3	-0,5
4					3	-0,5
5						
6						
7						
8						

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

**Risposta alla domanda 10 (4 pt):**