

# Programmazione I

## Prova scritta - 13 settembre 2018 - 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) 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++) f<<b[i] ; }
void leggi(int c[], int n){ ifstream f("numeri");
    for (int i = 0 ; i < n ; i++) f>>c[i]; }
main() {
    int c[] = {2, 3}; char b[] = {'6', ' ', '4'};
    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 4  
d) nessuna delle altre risposte è corretta
2. (3, -0.5) Supponendo che la matrice **int a[5][3]** sia memorizzata a partire dall'indirizzo 100 e che gli oggetti di tipo **int** occupino 4 byte, l'indirizzo dell'elemento **a[1][0]** è
- a)  $100 + 1*4*3 + 0*4 = 112$   
b)  $100 + 0*4*3 + 0*4 = 100$   
c)  $100 + 1*4*5 + 0*4 = 120$   
d) nessuna delle altre risposte è corretta

3. (2, -5) Il seguente programma:

```
int a = 10, b = 20 ;
int &fun(bool f) { if (f) return a ; else return b ; }
main() { int &c = fun(false) ; cout<<c; }
```

- a) Stampa 10  
b) Stampa 20  
c) Contiene un errore di gestione della memoria  
d) Nessuna delle altre risposte è corretta

4. (2, -.5) Il seguente programma:

```
int g = 10 ;  
void fun(int g) { cout<<(g++)<<" "; }  
main(){ fun(g) ; cout<<g ; }
```

- a) stampa 10 11
- b) stampa 10 10
- c) stampa 11 11
- d) stampa 11 10

## 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: -.5**
- **Una risposta lasciata in bianco viene calcolata: 0**

5. Indicare quale/i delle seguenti affermazioni sulle stringhe è/sono corrette;

- a) il terminatore non è indispensabile per le stringhe dinamiche
- b) il carattere '\n' non può essere utilizzato per indicare il fine stringa
- c) le stringhe dinamiche vengono memorizzate nello *heap*
- d) lo spazio occupato in memoria dall'*array* di caratteri utilizzato per rappresentare una stringa dipende dal numero corrente di caratteri della stringa

6. Data una sequenza di  $N$  valori interi positivi

- a) Se, in un dato programma, la sequenza è memorizzata in un *array* con un numero di elementi pari ad  $N$ , allora non c'è modo di conoscere tale numero di elementi in un punto del programma in cui è visibile l'*array* ma non è visibile alcuna variabile in cui è memorizzato il numero di elementi dell'*array* stesso
- b) Se la sequenza è memorizzata in una lista semplice, senza memorizzare in alcuna altra variabile il numero di elementi della sequenza, allora sono necessarie più di  $O(1)$  operazioni per calcolare tale numero di elementi
- c) Se la sequenza è memorizzata in un *array* utilizzando un terminatore per delimitare la fine della sequenza stessa, non è necessario aver memorizzato ulteriori informazioni in altre variabili per poter conoscere il numero di elementi nella sequenza
- d) Se le operazioni più frequenti sulla sequenza sono inserimenti ed estrazioni in/dal mezzo, allora la struttura dati più conveniente da utilizzare, in termini di costo computazionale, per implementare la sequenza è un lista

7. Dato il seguente programma e tenendo conto di possibili problemi di *overflow*:

```
main(){unsigned int a, b, c ;  
cin>>a>>b ; c = a - b ;  
if (c > 2)  
    cout<<"messaggio"<<endl ;}
```

- a) Sia **a** che **b** che **c** non possono mai assumere un valore maggiore del valore massimo consentito per il tipo **unsigned int**
- b) Per qualsiasi valore di **a** e **b**, **c** non assume un valore negativo dopo l'assegnamento **c = a - b ;**
- c) Il programma potrebbe non stampare **messaggio** anche se **c > 2** quando si esegue l'istruzione **if (c > 2)**
- d) Il programma potrebbe stampare **messaggio** anche se **a < b + 2** quando si esegue l'istruzione **c = a - b ;**

### 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. (7 pt) Descrivere la differenza tra output formattato ed output non formattato in non più di 10 righe, e rispondere alla seguente domanda “E’ possibile scrivere un file di testo utilizzando output non formattato, e se sì come?” (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).



9. (7 pt) Scrivere una funzione che prende in ingresso una stringa **S**, che si assume contenere la rappresentazione in base 2 di un numero **N**, e ritorna tale numero **N**. Ad esempio, se la stringa **S** è "101", la funzione ritorna 5. La lunghezza massima consentita per la stringa **S** è 10.



# Programmazione I

## Prova scritta - 13 settembre 2018

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					2	-0,5
4					2	-0,5
5						
6						
7						

**Risposta alla domanda 8 (7 pt):**

**Risposta alla domanda 9 (7 pt):**



# Programmazione I

## Prova scritta - 13 settembre 2018

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					2	-0,5
4					2	-0,5
5						
6						
7						

**Risposta alla domanda 8 (7 pt):**

**Risposta alla domanda 9 (7 pt):**