

# Programmazione I

## Prova scritta - 23 gennaio 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, -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

2. (3, -.5) Dato il seguente programma

```
int g = 1 ;  
double fun() { ++g ; return g + 3 ; }  
main()  
{ double m = 1 + fun() / 2 ;  
  cout<<static_cast<int>(g + m) ; }
```

- a) Se eseguito, il programma stampa 5
- b) Se eseguito, il programma stampa 4
- c) Il programma stampa un numero diverso da 5 e da 4
- d) Il programma può non stampare nulla, perché può essere terminato a causa di un errore di gestione della memoria

3. (3, -0.5) Supponendo che la matrice `int a[2][5]` sia memorizzata a partire dall'indirizzo 100 e che gli oggetti di tipo `int` occupino 4 byte, l'indirizzo dell'elemento

`a[1][2]` è

- a)  $100 + 0*4*5 + 2*4 = 108$
- b)  $100 + 1*4*2 + 1*4 = 112$
- c)  $100 + 1*4*5 + 2*4 = 128$
- 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: -.5
  - Una risposta lasciata in bianco viene calcolata: 0
4. Data una sequenza di  $N$  valori interi positivi
- a) La struttura dati più efficiente, in termini di occupazione di memoria, in cui si potrebbero memorizzare tali valori è una lista semplice di  $N$  elementi
  - b) Aggiungere un elemento nel mezzo della sequenza ha costo  $O(N)$  se la sequenza è implementata mediante un array
  - c) Accedere in lettura all' $i$ -esimo elemento della sequenza ha costo  $O(1)$  se la sequenza è implementata mediante una lista semplice e si conosce l'indirizzo dell'elemento in memoria
  - d) Se la sequenza è implementata mediante una lista semplice, allora eliminare l'elemento di indice  $N/2$  della sequenza ha costo  $O(N)$  anche se si conosce l'indirizzo dell'elemento in memoria
5. Data la rappresentazione dei numeri in virgola mobile mediante segno, mantissa ed esponente:
- a) Dato un numero in virgola mobile 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
6. Dato il seguente programma:
- ```
1:int a = 2;
2:
3:main()
4:{
5:    for (int i = 4 ; i > 2 ; i--) {
6:        int a = i / 2 ;
7:        cout<<(a * i)<<" ";
8:    }
9:    cout<<a;
10:}
```
- a) il programma stampa **8 3 1**
  - b) la variabile **a** definita alla riga 1 non è visibile in tutta la funzione **main**
  - c) il programma stampa **8 3 2**
  - d) la variabile **i** è visibile anche alla riga 9
7. Data l'istruzione **f>>i** ; ove **i** è una variabile di tipo **int** ed **f** è un oggetto di tipo **ifstream**, configurato per leggere numeri in base 10 ed associato ad un file correttamente aperto:
- a) Se, all'esecuzione di tale istruzione, non si è ancora raggiunta la fine del file, il programma legge i byte del file a partire dalla posizione corrente, alla ricerca della rappresentazione in base 10 di un numero intero
  - b) All'esecuzione di tale istruzione il programma si blocca sempre in attesa che vengano inseriti nuovi caratteri nel file associato all'*ifstream* **f**

- c) Se, all'esecuzione di tale istruzione, la porzione di file a partire dalla posizione corrente contiene una sequenza di caratteri che rappresenta un numero intero, seguita da uno spazio ed infine da una sequenza di caratteri che non rappresenta alcun numero, l'oggetto **f** non va in stato di errore
- d) Nel caso in cui tale istruzione sia eseguita con successo, ossia venga memorizzato correttamente un numero intero letto da **f** nella variabile **i**, eventuali caratteri successivi a quelli che rappresentano il numero letto vengono lasciati nell'*istream*

### 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 sintassi e semantica dell'operatore logico **&&**, descrivendo anche il corto circuito logico, ed illustrandolo con un esempio in cui se ne vedono le conseguenze. Non scrivere più di 10 righe, più un'eventuale tabella ed un eventuale frammento di codice. 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. In quanto agli eventuali esempi, si premierà la capacità di esporre i concetti mediante gli esempi più semplici possibili.



9. (7 pt) Scrivere una funzione che prende in ingresso un vettore di interi e lo modifica trasformandolo in un nuovo vettore che contiene gli stessi valori interi ordinati in senso crescente, tranne i valori che compaiono solo una volta nel vettore originale. Ad esempio, se la funzione prende in ingresso il vettore [4, 6, 3, 4, 1, 10, 6, 6], allora trasforma il vettore in [4, 6]. Il punteggio ottenuto è proporzionale all'efficienza della funzione implementata, in termini sia di numero di passi che di occupazione di memoria.



# Programmazione I

## Prova scritta - 23 gennaio 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/<br>Penalità |      |
|---|----------|---|---|---|--------------------|------|
|   | A        | B | C | D |                    |      |
| 1 |          |   |   |   | 3                  | -0,5 |
| 2 |          |   |   |   | 3                  | -0,5 |
| 3 |          |   |   |   | 3                  | -0,5 |
| 4 |          |   |   |   |                    |      |
| 5 |          |   |   |   |                    |      |
| 6 |          |   |   |   |                    |      |
| 7 |          |   |   |   |                    |      |

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

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

# Programmazione I

## Prova scritta - 23 gennaio 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/<br>Penalità |      |
|---|----------|---|---|---|--------------------|------|
|   | A        | B | C | D |                    |      |
| 1 |          |   |   |   | 3                  | -0,5 |
| 2 |          |   |   |   | 3                  | -0,5 |
| 3 |          |   |   |   | 3                  | -0,5 |
| 4 |          |   |   |   |                    |      |
| 5 |          |   |   |   |                    |      |
| 6 |          |   |   |   |                    |      |
| 7 |          |   |   |   |                    |      |

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

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