

Programmazione I

Prova scritta - 11 giugno 2019 - 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, -.5) Dato un programma C++ che contiene la definizione di una variabile **a** di un qualche tipo, e l'istruzione

```
cout<<a<<endl ;
```

- a) il codice da eseguire per la stampa del valore di **a**, in base al tipo di dato di **a**, viene stabilito a tempo di compilazione
- b) il codice da eseguire da eseguire per la stampa del valore di **a**, in base al tipo di dato di **a**, viene stabilito a tempo di esecuzione; ma è sempre lo stesso, indipendentemente dal valore memorizzato in **a**
- c) il codice da eseguire da eseguire per la stampa del valore di **a**, in base al tipo di dato di **a**, viene stabilito a tempo di esecuzione; e cambia in base al valore memorizzato in **a**
- d) nessuna delle altre risposte è vera

2. (3, -.5) Tenendo presente eventuali problemi di conversione (il tipo **unsigned int** è gerarchicamente superiore al tipo **int**) e di rappresentabilità, quale delle seguenti risposte è vera riguardo al seguente frammento di codice?

```
int delta_w; unsigned int wsum;
cin>>delta_w>>wsum;
if (wsum - delta_w < 0) cout<<"minore"<<endl ;
```

- a) Stampa **minore** se **delta_w** è positivo e strettamente maggiore di **wsum**
- b) Stampa minore solo in alcuni dei casi in cui il calcolo della differenza **wsum - delta_w** genera un *overflow*
- c) Non stampa nulla qualsiasi sia la coppia di valori interi inserita dall'utente
- d) Nessuna delle altre risposte è vera

3. (3, -0.5) Data la seguente funzione a cui viene passato un array **a** di dimensione **N**:

```
bool fun(unsigned int a[], int i, unsigned int N) {
    if ( ( i < 0 || i >= N ) || a[i] % 2 == 1 )
        return false ;
    return true ;
}
```

- a) Solo se l'indice **i** è compreso tra 0 ed **N-1** (estremi inclusi) la funzione controlla il valore dell'elemento **i**-esimo dell'array **a**, e ritorna **false** se tale elemento è pari
- b) Solo se l'indice **i** è compreso tra 0 ed **N-1** (estremi inclusi) la funzione controlla il valore dell'elemento **i**-esimo dell'array **a**, e ritorna **false** se tale elemento è dispari

- c) Nessuna delle altre risposte è vera
- d) La funzione può scrivere al di fuori dell'array a

4. (3, -.5) Assumendo che, nel seguente programma, 1) le scritture effettuate dalla funzione **scrivi** abbiano tutte successo, e 2) l'operazione di apertura del file in lettura nella funzione **leggi** abbia successo:

```
void scrivi(const int b[], int n){ ofstream f("numeri");
    for (int i = 0 ; i < n ; i++) f<<b[i] ; }
void leggi(int c[], int n){ ifstream f("numeri");
    f.read(reinterpret_cast<char *>(c), sizeof(int) * n); }
main() {
    int c[] = {2, 3}; int b[] = {4, 5};
    scrivi(b, 2) ; 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 4 5
- 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
5. Assumendo che i valori di tipo **double** siano rappresentati, mediante lo standard IEEE 754 in base 2, con una precisione di 53 cifre in base 2:
- a) Un qualsiasi numero reale che sia strettamente maggiore di zero e minore di 1, e che, nella sua rappresentazione in base 2, abbia un numero di cifre dopo la virgola inferiore a 53, è rappresentabile in modo esatto mediante il tipo **double**.
 - b) Un qualsiasi numero reale che sia strettamente maggiore di zero e minore di 1, e che, nella sua rappresentazione in base 10, abbia una sola cifra dopo la virgola, è rappresentabile in modo esatto mediante il tipo **double**.
 - c) Mediante il tipo di dato **double** non si possono rappresentare in modo esatto tutti i numeri reali compresi nell'intervallo di rappresentabilità del tipo di dato stesso.
 - d) Esistono numeri strettamente maggiori di zero e minori di 1, e con più di 53 cifre dopo la virgola nella loro rappresentazione in base 2, che si possono rappresentare in modo esatto mediante il tipo **double**
6. Dato il seguente programma
- ```
int g = 1 ;

int fun(char &g)
{ return 2 * (++g); }

main()
{
 char m = 1 ; char n = static_cast<char>(2.1) + fun(m) ;
 cout<<static_cast<int>(m * n) ;
}
```

- a) Se eseguito, il programma stampa **12**
- b) Nella inizializzazione della variabile **n**, la conversione `static_cast<char>(2.1)` causa perdita di informazione
- c) Nell'ultima istruzione del **main**, la conversione esplicita causa perdita di informazione
- d) Quando invocata nel **main**, la funzione **fun** non modifica la variabile globale **g**

7. 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

8. (6 pt) In non più di 10 righe di testo, più una eventuale tabella, 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 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 prenda in ingresso una stringa, che si assume contenere solo lettere minuscole, e ritorni vero se, la stringa contenga uno stesso carattere al massimo due volte, falso altrimenti. Ad esempio, la funzione ritorna vero per la stringa “*ippocastano*” e falso per la stringa “*pippo*”. Si ottiene il punteggio massimo se si realizza una funzione che scorre sia la stringa che altri eventuali array di appoggio al più una volta.



**Programmazione I**  
**Prova scritta – 11 giugno 2019**

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 |          |   |   |   | 2                  | -0,5 |
| 2 |          |   |   |   | 3                  | -0,5 |
| 3 |          |   |   |   | 3                  | -0,5 |
| 4 |          |   |   |   | 3                  | -0,5 |
| 5 |          |   |   |   |                    |      |
| 6 |          |   |   |   |                    |      |
| 7 |          |   |   |   |                    |      |

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

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

**Programmazione I**  
**Prova scritta - 11 giugno 2019**

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 |          |   |   |   | 3                  | -0,5 |
| 5 |          |   |   |   |                    |      |
| 6 |          |   |   |   |                    |      |
| 7 |          |   |   |   |                    |      |

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

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