

# Programmazione I

## Prova scritta - 8 giugno 2016 - 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) Supponendo che dopo aver lanciato il seguente programma l'utente immetta su *stdin* 123 e prema invio (e ricordando che l'oggetto *cin* di default salta gli spazi bianchi)

```
main()
{ char c ; int i = 0; cin>>c>>i ; cout<<c<<" "<<i ;
  cin>>c ; cout<<" "<<c ; }
```

- a) Il programma stampa 1 23, poi si blocca in attesa del prossimo invio di caratteri su *stdin*
- b) Il programma stampa 1 2 3, poi si blocca in attesa del prossimo invio di caratteri su *stdin*
- c) Il programma stampa 1 0 2
- d) Nessuna delle altre risposte è corretta

2. (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

3. (3, -.5) La seguente istruzione:

```
cout<<static_cast<int>('c' - 'a' + '4' - '1') ;
```

- a) può causare problemi di *overflow*
- b) immette il codice di un carattere non stampabile su *stdout*
- c) stampa 5
- d) nessuna delle altre risposte è corretta

4. (2, -0.5) Dato il seguente programma:

```
void fun(int &) ;  
int globale = 2;  
main() { globale = 3; fun(globale) ;}  
void fun(int &rif_loc) { rif_loc++; cout<<rif_loc<<" "<<globale ; }
```

- Se eseguito, il programma stampa 4 3
- Se eseguito, il programma stampa 4 4
- Nessuna delle altre risposte è vera
- Se eseguito, il programma stampa 4 2

## 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 quali delle seguenti affermazioni sono vere

- Il linguaggio macchina permette di scrivere programmi portabili tra diverse architetture
- Il linguaggio macchina non è un linguaggio di alto livello
- Un programma in linguaggio macchina è una sequenza di byte da non interpretarsi come sequenza di caratteri
- A parità di algoritmo implementato, l'esecuzione di un programma scritto in un linguaggio compilato (e quindi poi compilato) è tipicamente più veloce dell'esecuzione di un programma scritto in un linguaggio interpretato

6. Dato un puntatore **int \*p**, che punta ad un *array* di **int** allocato dinamicamente in memoria, l'istruzione: **p=0** ;

- fa sì che **p** non punti più ad alcun *array* in memoria dinamica;
- anche se **p** non era l'unico riferimento alla zona di memoria, impedisce di eliminare successivamente l'oggetto dalla memoria;
- non elimina l'*array* dalla memoria dinamica;
- modifica il valore di altri puntatori che puntano allo stesso *array*.

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

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

8. Dato il seguente programma:

```
const int N = 1000 ;  
struct ss { int b; char a[N] ; } ;  
void fun(ss d) { d.a[N/2] = 15 ; d.b = 4 ; cout<<d.b ; }  
main(){ ss c ; c.b = 10 ; c.a[N/2] = 2 ;  
    fun(c) ; cout<<" "<<c.b<<" "<<static_cast<int>(c.a[N/2]) ; }
```

- Quando **fun** è invocata il contenuto del parametro attuale **c** è copiato nel parametro formale **d**
- In quanto al campo **a** del parametro attuale **c** passato alla funzione **fun**, l'*array* in esso contenuto è copiato nel campo **a** del parametro formale **d**.
- Un passaggio per riferimento sarebbe stato più conveniente in termini di tempo di esecuzione
- Il programma stampa: 4 10 2



### 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) Descrivere sintassi, semantica e vantaggi del tipo *booleano*, in non più di otto righe, più eventuali frammenti di codice o ulteriori righe scritte in una qualsiasi notazione (a supporto della descrizione della sintassi).



10. **(5 pt)** Scrivere una funzione che prenda in ingresso un vettore di interi con un numero pari di elementi, e ritorni l'indice dell'elemento  $v$  con la seguente proprietà: il numero di elementi del vettore con valore minore o uguale a  $v$  (quindi incluso  $v$  stesso), è uguale al numero di elementi del vettore con valore strettamente maggiore di  $v$ . Ad esempio, dato il vettore  $[2, 5, 8, 4, 7, 1]$ , la funzione deve ritornare l'indice 3 (ossia l'indice dell'elemento di valore 4). Se c'è più di un elemento con tale proprietà, la funzione può ritornare uno qualsiasi di tali elementi.



# Programmazione I

## Prova scritta - 8 giugno 2016

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					2	-0,5
5						
6						
7						
8						

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



**Risposta alla domanda 10 (5 pt):**

# Programmazione I

## Prova scritta - 8 giugno 2016

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					2	-0,5
5						
6						
7						
8						

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

**Risposta alla domanda 10 (5 pt):**