

# Programmazione I

## Prova scritta - 10 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) 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. (2, -0,5) La memoria occupata da un oggetto **X** allocato dinamicamente all'interno di una funzione

- a) è allocata all'inizio dell'esecuzione della funzione;
- b) è liberata se si assegna l'indirizzo di un nuovo oggetto dinamico al puntatore che contiene l'indirizzo di **X** in memoria;
- c) è liberata se viene deallocato il puntatore che contiene l'indirizzo di **X** in memoria;
- d) se non esplicitamente deallocata, è liberata solo al termine del programma.

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

```
main ()
{
    int *a = new int[2]; a[0] = 1; a[1] = 2;
    int *b = a; b[0] = 3;
    cout<<a[0]<<" "<<a[1]<<" "<<b[0]<<endl ;
}
```

- a) Il programma stampa 1 2 3 e termina correttamente
- b) Il programma stampa 3 2 3 e termina correttamente
- c) Il programma potrebbe essere terminato a causa di un errore di gestione della memoria
- d) Nessuna delle altre risposte è vera

4. (3, -0.5) Dato il seguente programma:

```
int global ;
void fun(int &local){ local-- ; cout<<local<<" "<<global ; }
main() { int &ref = global; ref = 2 ; fun(ref) ; }
```

- Se eseguito, il programma stampa 1 2
- Se eseguito, il programma stampa 1 1
- Nessuna delle altre risposte è vera
- Il programma contiene un errore di gestione della memoria

## 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. Dato un qualsiasi programma in C/C++:

- Il compilatore decide la posizione degli oggetti con classe di memorizzazione statica, all'interno dello spazio di indirizzamento del processo corrispondente a tale programma
- Il compilatore decide la posizione degli oggetti con classe di memorizzazione dinamica, all'interno dei record di attivazione delle funzioni in cui tali oggetti sono allocati
- Il compilatore decide la posizione degli oggetti con classe di memorizzazione automatica, all'interno dei record di attivazione delle funzioni in cui tali oggetti sono definiti
- Il tempo necessario per eseguire il passaggio di un array ad una funzione è indipendente dalle dimensioni dell'array

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

- il terminatore di stringa non è indispensabile nel caso di un'allocazione dinamica della stringa
- il carattere '\n' non può essere utilizzato per indicare il fine stringa
- le stringhe statiche non 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

7. Dato il seguente programma:

```
const int N = 10 ; struct ss { char a[N] ; } ;
struct ss fun(ss & e) { e.a[0] = 'z' ; return e ;}
main()
{   ss c, d ; c.a[0] = 'y' ;
    d = fun(c) ; d.a[0] = 'k' ; cout<<c.a[0] ; }
```

- Il programma stampa **k**
- Il programma stampa **z**
- Il ritorno di **e** nella funzione **fun** comporta la copia del contenuto del parametro attuale a cui si riferisce il parametro formale **e**
- Quando **fun** è eseguita, il parametro formale **e** diventa un sinonimo per la variabile locale **c**

## 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 sintassi, semantica e motivazione del costrutto **for** in non più di 12 righe, utilizzando anche uno schema se lo si ritiene opportuno (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. (6 pt) Scrivere una funzione che prende in ingresso un vettore di valori interi, e ritorna vero solo se il vettore è costituito da una sequenza di coppie di valori uguali. Ad esempio, la funzione ritorna vero se il vettore è [2, 2, 4, 4, 3, 3], mentre ritorna falso se il vettore è, ad esempio, [2, 2, 2, 3, 3].



# Programmazione I

## Prova scritta - 10 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/ Penalità	
	A	B	C	D		
1					3	-0,5
2					2	-0,5
3					3	-0,5
4					3	-0,5
5						
6						
7						

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

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



# Programmazione I

## Prova scritta - 10 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/ Penalità	
	A	B	C	D		
1					3	-0,5
2					2	-0,5
3					3	-0,5
4					3	-0,5
5						
6						
7						

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

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