

# Programmazione I

## Prova scritta - 14 luglio 2017 - 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, -.5) Data la seguente stringa rappresentata mediante un array

```
char s[10] = "Soprano" ;
```

l'istruzione:

```
s[5] = ' ' ;
```

- a) contiene uno o più errori di accesso alla memoria
- b) trasforma la stringa da “Soprano” a “Sopra”
- c) non modifica la lunghezza della stringa
- d) nessuna delle altre risposte è vera

3. (2, -0.5) Contrassegnare l'affermazione corretta riguardo gli array:

- a) E' impossibile definire un array contenente oggetti di tipo eterogeneo
- b) I valori degli indici degli elementi partono da 1
- c) Se un array non viene inizializzato, ai suoi elementi vengono assegnati sempre valori casuali
- 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. Dato il seguente programma:

```
void inizializza(int a[][3], int N)
{ for(int i = 0 ; i < N ; i++)
    for(int j = 0 ; j < 3 ; j++)
        a[i][j] = 1 ; }
main() { int c[2][3] ; inizializza(c, 2) ; }
```

- a) La matrice *c* è vuota, ossia i suoi elementi non hanno nessun valore, quando viene passata alla funzione *inizializza*, che poi provvederà a dare un valore a ciascun elemento
- b) In quanto all'accesso agli elementi della matrice *a* all'interno della funzione *inizializza*, conoscere il numero di colonne della matrice è necessario per conoscere l'indirizzo in memoria di ciascun elemento della matrice
- c) In quanto all'accesso agli elementi della matrice *a* all'interno della funzione *inizializza*, conoscere il numero di righe della matrice è necessario per conoscere l'indirizzo in memoria di ciascun elemento della matrice
- d) Dopo l'invocazione della funzione *inizializza*, tutti gli elementi della matrice *c* contengono il valore 1

5. 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

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

- a) fa sì che **p** non punti più all'array;
- b) se **p** era l'unico puntatore all'array, impedisce di eliminare successivamente l'oggetto dalla memoria;
- c) fa sì che la memoria allocata per l'array sia liberata in quanto non più utilizzata;
- d) tenta di spostare l'array puntato da **p** all'indirizzo zero in memoria.

7. Data una sequenza di  $N$  valori interi positivi
- La struttura dati più efficiente, in termini di occupazione di memoria, in cui si potrebbero memorizzare tali valori è un *array* di  $N$  elementi
  - 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
  - Aggiungere un elemento in testa alla sequenza ha costo  $O(1)$  se la sequenza è implementata mediante una lista semplice
  - Se: 1) la sequenza è memorizzata in un array di dimensioni maggiori di  $N$ , e 2) si utilizza un terminatore per delimitare la fine della sequenza all'interno dell'array stesso, senza memorizzare in alcuna altra variabile aggiuntiva il numero di elementi della sequenza, allora sono necessarie più di  $O(1)$  operazioni per calcolare tale numero di elementi
8. Dato un programma scritto in linguaggio C/C++
- Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni dei parametri formali
  - Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili globali a cui accede la funzione
  - Per deallocare un record di attivazione è necessario reinizializzare il contenuto delle celle di memoria precedentemente occupate dal record stesso
  - Il record di attivazione di una funzione non contiene nessuno degli oggetti dinamici a cui la funzione può accedere

### 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 l'idea alla base e lo schema generale della rappresentazione in virgola mobile dei numeri reali, in non più di otto righe più eventuali schemi (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).



10. (6 pt) Scrivere una funzione che prenda in ingresso un vettore  $\mathbf{v}$  di interi (qualsiasi) ed un valore intero  $n$ , e rimuova dal vettore  $\mathbf{v}$  la più grande sotto-sequenza di elementi adiacenti con le seguenti caratteristiche: il primo elemento della sotto-sequenza ha indice  $n$ , tutti gli elementi della sotto-sequenza sono ordinati per valori crescenti. Ad esempio, presi in ingresso il vettore  $[2, 1, 5, 10, 9, 12]$  ed il valore 1, la funzione trasforma il vettore in  $[2, 9, 12]$ . Si ottiene il punteggio massimo se si implementa un algoritmo di complessità lineare e che non utilizza ulteriori vettori di appoggio.



# Programmazione I

## Prova scritta - 14 luglio 2017

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						
5						
6						
7						
8						

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

**Risposta alla domanda 10 (6 pt):**



# Programmazione I

## Prova scritta - 14 luglio 2017

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						
5						
6						
7						
8						

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

**Risposta alla domanda 10 (6 pt):**