

# Programmazione I

## Prova scritta - 25 gennaio 2012 - 1h10min

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) Quale delle seguenti affermazioni è vera?

- a) Un file binario può contenere sequenze di byte uguali a quelle che può contenere un file di testo
- b) Un file di testo non contiene numeri, ma rappresentazioni grafiche di caratteri (alcuni dei quali possono eventualmente essere cifre numeriche)
- c) Nessuna delle altre risposte è vera
- d) Un file binario utilizzato per memorizzare un *array* di caratteri non contiene alcuna sequenza di byte rappresentanti caratteri

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

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

- a) Se eseguito, il programma stampa 4 3
- b) Se eseguito, il programma stampa 4 4
- c) Nessuna delle altre risposte è vera
- d) Se eseguito, il programma stampa 4 2

3. (2, -0.5) Data la seguente funzione a cui viene passato un array **a** di dimensione **N**, e facendo attenzione ad eventuali errori di gestione della memoria:

```
bool fun(unsigned int a[], int i, unsigned int N) {
    if (a[i] % 2 == 1 || (i < 0 && i >= N))
        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 e ritorna **true** 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 e ritorna **true** se tale elemento è dispari
- c) Nessuna delle altre risposte è vera
- d) Può accadere che la funzione scriva al di fuori dell'array **a**

4. (2, -0.5) Data la seguente stringa rappresentata mediante un array

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

le istruzioni:

```
s[2] = 't' ; s[3] = s[2] ; s[4] = 'o' ; s[5] = '\\0' ;
```

- a) contengono uno o più errori di accesso alla memoria
- b) trasformano la stringa da “Soprano” a “Sotto”
- c) non modificano la lunghezza della stringa
- d) nessuna delle altre risposte è vera

## 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: -0.5
- Una risposta lasciata in bianco viene calcolata: 0

5. Quali delle seguenti affermazioni sono vere?

- a) In C non esiste il passaggio di un oggetto per riferimento, ma lo si può emulare passando l'indirizzo dell'oggetto stesso
- b) La funzione **printf** della libreria standard del C ha un numero di argomenti variabile
- c) Per scrivere in una variabile un valore letto da *stdin* usando la funzione **scanf** della libreria standard del C, è necessario passare alla funzione sia l'indirizzo che il tipo della variabile in cui scrivere il valore letto
- d) Nella funzione **scanf** della libreria standard del C, sbagliare il tipo o l'indirizzo della variabile in cui si desidera scrivere il valore letto da *stdin* può causare corruzione della memoria

6. Data la seguente definizione:

```
char c = static_cast<int>('b') ;
```

- a) all'atto dell'inizializzazione si ha perdita di informazione
- b) si inizializza **c** con il valore della costante carattere **'b'**
- c) il compilatore deve introdurre una conversione implicita per realizzare l'inizializzazione
- d) se si tentasse di stampare il valore di **c** mediante **cout<<c**; si stamperebbe il codice ASCII del carattere **b**

7. Definita una parola come una sequenza di caratteri non separati da spazi, il seguente programma:

```
struct parola {char *stringa ; int lun ;} ;  
main()  
{  
    parola v ; cin>>v.stringa ; v.lun = strlen(v.stringa) ;  
    cout<<v.stringa<<" "<<v.lun<<endl ;  
}
```

- a) legge correttamente da *stdin* una parola e stampa su *stdout* tale parola seguita dalla rispettiva lunghezza
- b) contiene un errore di gestione della memoria
- c) nessuna delle altre risposte è vera
- d) potrebbe essere terminato forzatamente prima di aver stampato la parola

8. Nel seguente programma:

```
1:int a = 1;
2:
3:fun()
4:{
5:    int i ;
6:    for (i = 0 ; i < 5 ; i++) {
7:        float a = i / 2.0 ;
8:        cout<<a * i ;
9:    }
10:    cout<<a + i ;
11:}
12:
13:main()
14:{
15:    float b = 1.4 ;
16:    fun() ;
17:    cout<<a + b<<endl ;
18:}
```

- a) la variabile **b** definita alla riga 15 ha tempo di vita pari all'intero programma;
  - b) la variabile **a** definita alla riga 1 non ha visibilità pari all'intero programma;
  - c) la variabile **a** definita alla riga 1 non è più visibile alla riga 17
  - d) la variabile **a** definita alla riga 1 è visibile alla riga 10.
9. Durante l'esecuzione del programma in linguaggio macchina ottenuto dalla compilazione di un programma scritto in linguaggio C/C++
- a) Se in un dato istante non vi è più memoria sufficiente per creare un nuovo record di attivazione viene liberata memoria dinamica per fargli posto
  - b) In ogni istante le dimensioni dello *stack* sono proporzionali al numero di funzioni invocate e non ancora terminate
  - c) La scrittura al di fuori di un *array* locale ad una funzione non comporta l'aumento delle dimensioni del record di attivazione della funzione
  - d) Il valore iniziale di una variabile non inizializzata, e locale ad una funzione diversa dal *main*, dipende solo dai valori memorizzati nella sequenza di celle di memoria riservate al programma al suo avvio

### 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

10. (4 pt.) Cosa scrive su *stdout* il seguente programma?

```
void fun(char *s, char c)
{
    int i = 0;
    while (s[i] != '\0') {
        if (s[i] == c && s[i+1] != '\0')
            s[i+1] = c ;
        i++ ;
    }
}

main()
{
    char stringa[] = "ieri";
    cout<<stringa<<endl ; fun(stringa, 'e'); cout<<stringa<<endl ;
}
```

11. **(5 pt)** Scrivere una funzione che prenda in ingresso un vettore di interi ed un valore intero  $n$ , e rimuova dal vettore la sequenza di elementi con le seguenti caratteristiche: il primo elemento della sequenza ha indice  $n$ , tutti gli elementi della 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]. Ogni situazione di errore va controllata all'interno della funzione.



# Programmazione I

## Prova scritta - 25 gennaio 2012

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_ Corso di Laurea: \_\_\_\_\_

**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 un 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					2	-0,5
4					2	-0,5
5						
6						
7						
8						
9						

**Risposta alla domanda 10 (4 pt):**

**Risposta alla domanda 11 (5 pt):**



# Programmazione I

## Prova scritta - 25 gennaio 2012

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_ Corso di Laurea: \_\_\_\_\_

**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 un 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					2	-0,5
4					2	-0,5
5						
6						
7						
8						
9						

**Risposta alla domanda 10 (4 pt):**

**Risposta alla domanda 11 (5 pt):**