

# Programmazione II

## Prova di autovalutazione – 25 Maggio 2015

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

Alle seguenti domande corrisponde una sola risposta corretta fra quelle elencate.

- 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,-0.5] In un vettore di bit, come si fa per controllare lo stato di più bit con una sola operazione, lasciando inalterati tutti gli altri bit?

- A) Operazione di AND con una maschera in cui solo i bit da controllare sono settati a 1
- B) Operazione di XOR con una maschera in cui solo i bit da controllare sono settati a 0
- C) Operazione di AND con una maschera in cui tutti i bit sono settati a 1
- D) Nessuna delle precedenti risposte è vera

2) [2, -0.5] Dato il file di testo di nome **pip**, cosa stampa il seguente frammento di script shell?

```
var1=pippo  
var2=var1  
echo var2
```

- A) La stringa "pippo"
- B) Il contenuto del file pip
- C) La stringa "var1"
- D) Nessuna delle precedenti risposte

3) [2, -0.5] Cosa stampa la seguente sequenza di comandi shell?

```
[ 2 -ge 2 ] ; echo $?
```

- A) Sempre 1
- B) Sempre 0
- C) Un valore diverso da 0, ma non necessariamente uguale a 1
- D) Nessuna delle precedenti risposte

4) [2, -0.5] Cosa stampa a video il seguente frammento di codice?

```
int funz(int a, int b)
{
    if (a % 2 == 1) return b;
    return funz(a/2, b/2);
}

int main()
{
    cout<< funz(16,47)<<endl ;
    return 0 ;
}
```

- A) Non è garantita la terminazione delle chiamate ricorsive
- B) Stampa 1
- C) Stampa un numero intero diverso da 1
- D) Nessuna delle precedenti risposte è corretta

5) [2, -0.5] Date le seguenti definizioni, quali sono i livelli di tracing attivati?

```
unsigned int MASK = 5;
#define DBG(A, B) {if ((A) & MASK) {B; } }
#define D1(a) DBG(1, a) //Livello 1
#define D2(a) DBG(2, a) //Livello 2
#define D3(a) DBG(3, a) //Livello 3
```

- A) Sono attivati solo i livelli 1 e 3
- B) E' attivato solo il livello 1
- C) E' attivato solo il livello 3
- D) Nessuna delle precedenti risposte è corretta

6) [2, -0.5] Cosa rappresenta il seguente oggetto?

```
int (*funz)[10];
```

- A) Un puntatore ad un array di dieci interi non inizializzati
- B) Un array di 10 puntatori ad int non inizializzati
- C) Un array di 10 puntatori a funzioni aventi valore di ritorno intero
- D) Nessuna delle risposte precedenti

7) [2, -0.5] Assumendo che il tipo di dato int venga rappresentato su 4 byte, cosa fa il seguente codice?

```
int main() {
    const int SIZE = 2;
    int arr[SIZE];
    int *p = arr;
    *(p+4)=100;
}
```

- A) Provoca un errore a tempo di compilazione
- B) Assegna il valore 100 a `arr[1]`
- C) Assegna il valore 100 a `arr[2]`
- D) Nessuna delle precedenti risposte è corretta

## PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE

Ognuna delle seguenti domande può avere una o più risposte CORRETTE. Ogni risposta esatta viene calcolata +1, ogni risposta errata -0.5, e ogni risposta lasciata in bianco 0.

8) Nel linguaggio C++, l'operatore `&` può venire usato:

- A) Per l'esecuzione in background di una funzione
- B) Per l'accesso all'indirizzo di memoria di un oggetto
- C) Per l'operazione di AND bit a bit
- D) Per la creazione di un riferimento

9) L'espressione `extern int n;`

- A) Non provoca allocazione di memoria
- B) Rappresenta una dichiarazione di una variabile intera
- C) Definisce la variabile `n` e la rende accessibile ad altri file (mediante ridichiarazione)
- D) Necessita di almeno un file in cui sia definita la variabile `n` con collegamento esterno

10) Indicare quali delle seguenti affermazioni è vera in caso di invocazione del comando `make` in presenza di un `Makefile` contenente la seguente regola:

```
target.o: myfile
    comando1
    comando2
```

- A) Se il target è più recente del file di nome `myfile`, i comandi devono essere rieseguiti
- B) Se esiste un target di nome `myfile` nel `Makefile`, le sue regole devono essere controllate, indipendentemente dalla sua data di ultima modifica
- C) Se non esiste alcun file di nome `target.o`, i comandi devono essere rieseguiti
- D) Le regole implicite di compilazione permettono di evitare di inserire tra i prerequisiti l'eventuale file sorgente di nome `target.cc`

11) Se si invoca il comando `ls` in un terminale, la shell (bash) cerca il corrispondente eseguibile:

- A) In tutte le cartelle indicate nella variabile d'ambiente `PATH`
- B) Nella cartella corrente
- C) Nella cartella `/usr/bin`
- D) Nella cartella home dell'utente

### 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.

- 12) [3 pt.] Descrivere brevemente i concetti di visibilità, collegamento interno e collegamento esterno di identificatori dichiarati in file sorgenti C++ nel contesto di programmi organizzati su più file.
- 13) [4 pt.] Scrivere una funzione `HeapSort(int A[])` che implementi l'algoritmo di ordinamento heapsort in senso decrescente su un dato array A di interi fornito in input. Si supponga che la dimensione dell'array sia contenuta in `A[0]` e che siano disponibili le seguenti funzioni (con indicizzazione degli elementi vista a lezione):
  1. `BuildMinHeap(int A[])` per la creazione di un min-heap da un array di input A non ordinato
  2. `MinHeapify(int A[],int n)` per il ripristino della proprietà di min-heap a partire dal nodo n nel caso in cui i sottoalberi destro e sinistro di n soddisfino la proprietà di min-heap

## Programmazione II

### Prova di autovalutazione – 25 Maggio 2015

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	x				2	-0,5
2				x	2	-0,5
3		x			2	-0,5
4			x		2	-0,5
5	x				2	-0,5
6	x				2	-0,5
7				x	2	-0,5
8		x	x	x		
9	x	x		x		
10		x	x	x		
11	x					

**Risposta alla domanda 12 (3 pt):**

**Risposta alla domanda 13 (4 pt):**

```
void HeapSort(int A[]) {
    int length = A[0];
    BuildMinHeap(A);
    for (int i = length; i > 1; i--) {
        int temp = A[1];
        A[1] = A[i];
        A[i] = temp;
        A[0]--;
        MinHeapify(A, 1);
    }
}
```