

Informatica B

Tema B

20 Novembre 2003

1. È severamente vietato usare, durante lo svolgimento del tema d'esame, altro materiale che non siano i fogli del compito ed una penna e/o matita.
2. I punti ottenibili sono al massimo 12. L'esercizio numero 2 vale più degli esercizi 1 e 3 ed il suo svolgimento è condizione necessaria per ottenere la sufficienza, fissata in punti 6.

Cognome:

Nome:

Matricola:

1 Esercizio 1

1. Immaginate di avere un calcolatore che invece di poter rappresentare i classici due valori per ogni bit (0/1) possa rappresentare 7 valori per ogni bit.
 - (a) Quanti bit gli occorreranno per rappresentare i numeri naturali fino a 300 ?

- (b) Quanti bit risparmia rispetto ad una macchina che invece usa la classica rappresentazione binaria?

2. Spiegare brevemente qual è la funzione del compilatore e la sua importanza per il programmatore.

3. Spiegare, magari con un esempio, il significato di ognuna delle fasi del ciclo fetch-decode-execute nella macchina di Von Neumann.

2 Esercizio 2

Si scrivano, nel linguaggio **C**, le dichiarazioni di tipo necessarie per rappresentare un punto in uno spazio tridimensionale usando la rappresentazione in coordinate cartesiane (si chiami **Punto** il tipo corrispondente) ed un insieme di, al massimo, 100 punti (si chiami **InsiemePunti** il tipo corrispondente).

Si scriva un programma in linguaggio **C** che, data una variabile di tipo **InsiemePunti**, permetta all'utente di immettere un numero grande a piacere di elementi di tipo **Punto** (ovviamente tale numero non può essere superiore a 100), e che, alla fine dell'immissione, stampi quanti, tra gli elementi di tipo **Punto** immessi, hanno la caratteristica di avere lo stesso valore delle tre coordinate cartesiane (si pensi ad esempio ad un punto con coordinate $x=3$ $y=3$ $z=3$) e quali di questi è il più lontano dall'origine degli assi.

3 Esercizio 3

Sia dato il seguente programma in C:

```
void main(){

    int x, y, z;
    int *p1, *p2, *p3;

    x=3;
    y=4;
    z=5;

    -----

    -----

    -----

    printf("%d %d %d %d %d %d \n",x,y,z,*p1,*p2,*p3);

    if (*p2 == 5) {

        *p1 = *p2;

    } else {

        p3=&x;

    }

    printf("%d %d %d %d %d %d \n",x,y,z,*p1,*p2,*p3);

}
```

1. Lo si completi scrivendo le istruzioni che fanno puntare `p1` a `y`, `p3` a `z` e `p2` a `x` e, dopo averlo completato, si dica qual è l'output del programma.

2. Se avessi un puntatore dichiarato come

```
int ** pp;
```

potrei assegnare a tale puntatore l'indirizzo di `p2`? e potrei assegnare, sempre a tale puntatore, il valore contenuto in `p3`? Giustificare brevemente le risposte (non saranno accettate risposte del tipo si/no)