

	Politecnico di Milano Facoltà di Ingegneria Industriale <b>INFORMATICA B</b> Prova in itinere del 13 Novembre 2012		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA
			Spazio riservato ai docenti <div style="border: 1px solid black; width: 100px; height: 20px; margin-left: auto;"></div>

- Il presente plico contiene 3 esercizi e deve essere debitamente compilato con cognome e nome, numero di matricola, posizione durante lo scritto (comunicata dal docente).
- Il tempo a disposizione è di 1 ora e 15 minuti.
- Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** (o ripudiate) con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **È possibile scrivere a matita** (e non occorre ricalcare al momento della consegna!) **facendo in modo comunque che quanto scritto sia ben leggibile**.
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- È ammessa la consultazione di **libri e appunti**, purché con pacata discrezione e senza disturbare.
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- È possibile **ritirarsi senza penalità**.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.

## Esercizio 1 (7 punti)

Si considerino le seguenti dichiarazioni di tipi e variabili che definiscono le strutture dati per rappresentare informazioni relative alla carriera accademica degli studenti di un'università:

```
#define MAXESAMI 30
typedef char stringa[15];

typedef struct {
    char codiceCorso[10];
    stringa nomeCorso;
    int voto;
} esame;

typedef struct{
    char matricola[8];
    stringa cognome;
    stringa nome;
    stringa corsoDiLaurea;
    int numEsamiSuperati;
    esame ElencoEsamiSuperati[MAXESAMI];
} studente;
```

1. Definire, usando il linguaggio C, un'appropriata variabile per memorizzare le informazioni relative a 1000 studenti. Si chiami tale variabile `InsiemeStudenti`.
2. Scrivere in linguaggio C, aggiungendo eventualmente opportune dichiarazioni di variabili, un frammento di codice che permetta di visualizzare a video, per ogni studente che ha una media voti superiore a 25, la matricola, nome, cognome, e la media voti. Si supponga che l'insieme degli studenti sia memorizzato nella variabile `InsiemeStudenti` definita al punto 1 e che essa sia già stata inizializzata con le informazioni relative a 1000 studenti.

## Soluzione

Punto 1

```
studente InsiemeStudenti[1000];
```

Punto 2

```
.....
void main()
{
    studente InsiemeStudenti[1000]; /* la stessa definizione di variabile del punto 1 riportata nel
                                     main*/

    int i;
    float sommaVoti;

    ....

    for (i=0; i<1000; i++)
    {
        sommaVoti =0;

        for (j=0; j<InsiemeStudenti[i].numEsamiSuperati; j++)
            sommaVoti = sommaVoti + InsiemeStudenti[i].ElencoEsamiSuperati[j].voto;

        if (sommaVoti / InsiemeStudenti[i].numEsamiSuperati > 25)
        {
            printf("Matricola:%s, Nome: %s, Cognome: %s, Media voti: %f \n",
```

```
InsiemeStudenti[i].matricola, InsiemeStudenti[i].nome,  
InsiemeStudenti[i].cognome,  
sommaVoti / InsiemeStudenti[i].numEsamiSuperati);
```

```
}
```

```
}
```

```
}
```

## Esercizio 2 (6 punti)

Si consideri il seguente codice, prestando attenzione ai tipi delle variabili:

```
#include <stdio.h>

void main() {
    int h, k, i, d2;
    float p, q, d1;

    printf("Inserisci due numeri: ");
    scanf("%d %d", &h, &k);
    d1=k;
    d2=k;
    p=1;
    q=1;
    for (i=0; i<k; i++) {
        p=p*(h/d1);
        q=q*(h/d2);
    }

    printf("p=%f q=%f\n", p ,q);
}
```

1. Si supponga di eseguire il programma 2 volte fornendo in ingresso le seguenti coppie di valori:

- 2 2
- 1 2

Per ogni esecuzione si dica, giustificando adeguatamente la risposta, qual è l'output del programma e in particolare quali valori conterranno le variabili **p** ed **q** alla fine dell'esecuzione.

2. Si consideri una generica esecuzione del programma. Si esprima il contenuto delle variabili **p** e **q** alla fine dell'esecuzione del programma in funzione dei valori forniti in ingresso dall'utente e memorizzati nelle variabili **h** e **k**.
3. Estendere il codice precedente in modo da memorizzare in un'apposita struttura dati/variabile la sequenza di valori assunti dalla variabile "**q**" durante l'esecuzione del codice. Aggiungere inoltre una porzione di codice dopo il ciclo for che, sfruttando la struttura dati/variabile definita in precedenza, stampi a video la sequenza di valori assunti dalla variabile **q** durante l'esecuzione. Si supponga che il valore di **k**, fornito in ingresso dall'utente, sia al massimo pari a 100.

## Soluzione

Punto 1

Dati in ingresso 2 e 2.

Valori delle variabili alla fine dell'esecuzione:

$p=1$ ,  $q=1$

Dati in ingresso 1 e 2.

Valori delle variabili alla fine dell'esecuzione:

$p=0.25$ ,  $q=0$

Si noti che l'espressione  $h/d1$  comporta l'esecuzione di una divisione tra float mentre l'espressione  $h/d2$  comporta l'esecuzione di una divisione tra interi.

Punto 2

Alla fine dell'esecuzione la variabile  $p$  sarà pari a  $(h/k)^k$

Alla fine dell'esecuzione la variabile  $q$  sarà pari a  $(parte\_intera(h/k))^k$

Punto 3

```

#include <stdio.h>

void main() {
    int h, k, i, d2;
    float p, q, d1;
    float seqValoriq[100];

    printf("Inserisci due numeri: ");
    scanf("%d %d", &h, &k);
    d1=k;
    d2=k;
    p=1;
    q=1;
    for (i=0; i<k; i++) {
        p=p/(h/d1);
        q=q*(h/d2);
        seqValoriq[i]=q;
    }

    printf("p=%f q=%f\n", p ,q);

    for (i=0; i<k; i++) {
        printf("%f ",seqValoriq[i]);
    }
}

```

### Esercizio 3 (4 punti)

1. Si determini la codifica binaria dei numeri 6.4 e -10.25 secondo lo Standard IEEE 754-1985 a precisione singola, riportando i calcoli effettuati.
2. L'uso della rappresentazione IEEE 754-1985 a precisione singola ha comportato delle approssimazioni?
3. L'uso della rappresentazione IEEE 754-1985 a precisione doppia avrebbe portato ad una rappresentazione più precisa dei due numeri considerati?

#### NOTA!

Si ricorda che lo standard IEEE 754-1985 a precisione singola ha le seguenti caratteristiche:

- 1 bit per il segno, 23 bit per la mantissa, 8 per l'esponente (K=127)

Si ricorda che lo standard IEEE 754-1985 a precisione doppia invece ha le seguenti caratteristiche:

- 1 bit per il segno, 52 bit per la mantissa, 11 per l'esponente (K=1023)

#### Soluzione

Punto 1

La codifica di 6.4 in precisione singola è

S=0

E=10000001

M=10011001100110011001100

La codifica di -10.25 in precisione singola è

S=1

E=10000010

M=010010000000000000000000

Punto 2

La rappresentazione in precisione singola del numero 6.4 comporta un'approssimazione.

La rappresentazione in precisione singola del numero -10.25 è esatta.

Punto 3

Essendo il numero -10.25 rappresentabile senza approssimazioni usando la precisione singola, l'uso della precisione doppia non ha nessun impatto sulla rappresentazione di tale numero. In entrambi i casi la rappresentazione è esatta.

L'uso della rappresentazione in precisione doppia del numero 6.4 invece permette di ottenere una rappresentazione più precisa di tale numero avendo a disposizione più bit. Comunque anche usando la precisione doppia vi è ancora un'approssimazione nella rappresentazione del numero 6.4.