

	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 alle tessere fedeltà dei clienti di una compagnia aerea:

```
#define MAXVIAGGI 100
typedef char stringa[15];

typedef struct {
    stringa aeroportoPartenza;
    stringa aeroportoArrivo;
    float distanza;      /* La distanza è espressa in chilometri e rappresenta la lunghezza del volo */
} viaggio;

typedef struct{
    char codiceTessera[10];
    stringa nome;
    stringa cognome;
    stringa nazionalita;
    int numViaggiEffettuati;
    viaggio ElencoViaggi[MAXVIAGGI];
} cliente;
```

1. Definire, usando il linguaggio C, un'appropriata variabile per memorizzare le informazioni relative a 50 clienti. Si chiami tale variabile ElencoClienti.
2. Scrivere in linguaggio C, aggiungendo eventualmente opportune dichiarazioni di variabili, un frammento di codice che permetta di visualizzare a video, per ogni cliente che ha effettuato almeno 10 viaggi, il nome, cognome, numero totale di chilometri percorsi e la lunghezza media dei voli. Si supponga che l'elenco dei clienti sia memorizzato nella variabile ElencoClienti definita al punto 1 e che essa sia già stata inizializzata con le informazioni relative a 50 clienti.

## Soluzione

Punto 1

```
cliente ElencoClienti[50];
```

Punto 2

```
.....
void main()
{
    cliente ElencoClienti[50]; /* la stessa definizione di variabile del punto 1 riportata nel main*/
    int i;
    float totKM;

    .....

    for (i=0; i<50; i++)
    {
        if (ElencoClienti[i].numViaggiEffettuati>=10)
        {
            totKM=0;

            for (j=0;j<ElencoClienti[i].numViaggiEffettuati;j++)
                totKM = totKM + ElencoClienti[i].ElencoViaggi[j].distanza;

            printf("Nome: %s, Cognome: %s, TotKM: %f, Lunghezza media voli: %f\n",
                ElencoClienti[i].nome, ElencoClienti[i].cognome,
```

```
totKM, totKM/ElencoClienti[i].numViaggiEffettuati);
```

```
}
```

```
}
```

```
}
```

## Esercizio 2 (6 punti)

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

```
#include <stdio.h>

void main() {
    int a,b,c1;
    float c2,d=0,f=0;

    printf("Inserisci due numeri: ");
    scanf("%d %d", &a, &b);
    c1=b;
    c2=b;
    while (b>0) {
        d=d+(a/c1);
        f=f+(a/c2);
        b=b-1;
    }

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

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

- 2 2
- 5 10

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

2. Si consideri una generica esecuzione del programma. Si esprima il contenuto delle variabili **d** e **f** alla fine dell'esecuzione del programma in funzione dei valori forniti in ingresso dall'utente e memorizzati nelle variabili a e b.
3. Si estenda il codice precedente in modo da memorizzare in un'apposita variabile di tipo array tutta la sequenza di valori assunti dalla variabile **d** durante l'esecuzione del codice, supponendo che il valore di **b**, fornito in ingresso dall'utente, sia al massimo pari a 50. Aggiungere inoltre una porzione di codice alla fine del programma che, sfruttando la variabile di tipo array definita in precedenza, stampi a video la sequenza di valori assunti dalla variabile **d** durante l'esecuzione.

## Soluzione

Punto 1

Dati in ingresso 2 e 2.

Valori delle variabili alla fine dell'esecuzione:

d=2, f=2

Dati in ingresso 5 e 10.

Valori delle variabili alla fine dell'esecuzione:

d=0, f=5

Si noti che il risultato ottenuto del calcolo in d e f è diverso perché, pur contenendo le variabili coinvolte gli stessi valori, la divisione eseguita per ottenere il valore di d è una divisione tra interi mentre quella eseguita per ottenere il valore di f è una divisione tra float.

Punto 2

Alla fine dell'esecuzione la variabile d sarà pari a  $b \cdot \text{parte\_intera}(a/b)$

Alla fine dell'esecuzione la variabile f sarà pari a  $b \cdot (a/b)$ , ossia a.

Punto 3

```

#include <stdio.h>

void main() {
    int a,b,c1;
    float c2,d=0,f=0;
    float seqValorid[50];
    int i;

    printf("Inserisci due numeri: ");
    scanf("%d %d", &a, &b);
    c1=b;
    c2=b;
    i=0;
    while (b>0) {
        d=d+(a/c1);
        seqValorid[i]=d;
        i=i+1;
        f=f+(a/c2);
        b=b-1;
    }

    printf("d=%f f=%f\n", d,f);

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

```

### Esercizio 3 (4 punti)

1. Si determini la codifica binaria dei numeri -12.625 e 16.65 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 -12.625 in precisione singola è

S=1

E=10000010

M=1001010000000000000000

La codifica di 16.65 in precisione singola è

S=0

E=10000011

M=00001010011001100110011

Punto 2

La rappresentazione in precisione singola del numero -12.625 è esatta.

La rappresentazione in precisione singola del numero 16.65 comporta un'approssimazione.

Punto 3

Essendo il numero -12.625 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 16.15 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 16.15.