

	Politecnico di Milano Facoltà di Ingegneria Industriale <b>INFORMATICA B</b> Prova in itinere del 16 Novembre 2011		COGNOME E NOME				
	RIGA	COLONNA	MATRICOLA				
			<i>Spazio riservato ai docenti</i>				
			<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>				

- Il presente plico contiene 4 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 2 ore.
- 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 ricaricare al momento della consegna!).
- 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 (5 punti)

1. Si definisca il tipo **Mammifero**, che permetta di rappresentare informazioni relative ai mammiferi ospitati in un parco protetto. In particolare, per ogni mammifero è necessario rappresentare i seguenti dati:
  - **Nome tipo**: una sequenza di 15 caratteri
  - **Tipo di alimentazione**: può essere carnivoro, onnivoro, erbivoro
  - **PesoMedio**: un float
  - L'insieme di **Esemplari** (al massimo 100); ogni esemplare è un dato a sua volta strutturato, composto da:
    - **CodiceEsemplare**: un intero
    - **AnnoDiNascita**: un intero
    - **Peso**: un float
  - **Numero di esemplari presenti nel parco**: un intero
2. Si dichiari il tipo **archMammiferi**, in grado di rappresentare l'insieme dei mammiferi presenti nel parco (si ipotizzi che ci siano al massimo 30 tipi diversi di mammiferi).
3. Si dichiari la variabile **mammiferi** di tipo **archMammiferi**
4. Si scriva in linguaggio C, aggiungendo eventualmente opportune dichiarazioni di variabili, il frammento di codice che costruisce l'archivio dei mammiferi sotto peso. Un tipo di mammifero si dice sotto peso se la media dei pesi di tutti i suoi esemplari è minore del peso medio definito per quella specie.

## Soluzione

```
1.
typedef enum {CARNIVORO, ONNIVORO, ERBIVORO} Alimentazione;

typedef struct{
    int codiceEsemplare;
    int annoNascita;
    float peso;
} Esemplare;

typedef struct{
    char nome[15];
    Alimentazione alim;
    float pesoMedio;
    Esemplare esemplari[100];
    int numEsemplari;
} Mammifero;

2.
typedef Mammifero archMammiferi[30]

3.
archMammiferi mammiferi;

4.
// Variabili aggiuntive
int numMammiferi; // Numero di elementi nell'array archMammiferi
Mammifero archMammiferiSottoPeso[30];
int numMammiferiSottoPeso = 0;
float pesoMedioEsemplari;
```

```
for (i = 0; i < numMammiferi; i++)
{
    pesoMedioEsemplari = 0;
    for(j=0; j<mammiferi[i].numEsemplari; j++)
        pesoMedioEsemplari = pesoMedioEsemplari + mammiferi[i].esemplari[j].peso;
    if(pesoMedioEsemplari/mammiferi[i].numEsemplari < mammiferi[i].pesoMedio)
    {
        archMammiferiSottoPeso[numMammiferiSottoPeso] = mammiferi[i];
        numMammiferiSottoPeso++;
    }
}
```

## Esercizio 2 (4 punti)

Scrivere un programma in linguaggio C che esegua le seguenti operazioni:

- Legge da tastiera due valori positivi minori di 100 che vengono memorizzati in due variabili intere  $m$  e  $n$
- Acquisisce, sempre da tastiera,  $m \times n$  valori interi che memorizza in una matrice  $\mathbf{A}$  di dimensioni massime  $100 \times 100$
- Costruisce una matrice  $\mathbf{P}$  di valori interi contenente solo le colonne con indice pari della matrice  $\mathbf{A}$  e una matrice  $\mathbf{D}$  di valori interi contenente solo le colonne con indice dispari della matrice  $\mathbf{A}$
- Infine, se le matrici  $\mathbf{P}$  e  $\mathbf{D}$  hanno le stesse dimensioni, calcola anche la matrice  $\mathbf{B} = \mathbf{P} - \mathbf{D}$

**Note:** Il valore 0 è un numero pari. Non è necessario stampare a video le matrici  $\mathbf{P}$ ,  $\mathbf{D}$  e  $\mathbf{B}$ .

### Esempio

$m = 3$   $n = 4$

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ -5 & -6 & -7 & -8 \\ 9 & 20 & 21 & 15 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 3 \\ -5 & -7 \\ 9 & 21 \end{bmatrix} \quad D = \begin{bmatrix} 2 & 4 \\ -6 & -8 \\ 20 & 15 \end{bmatrix} \quad B = \begin{bmatrix} -1 & -1 \\ 1 & 1 \\ -11 & 6 \end{bmatrix}$$

### Soluzione

```
#include <stdio.h>
#define S 100

void main() {
    int A[S][S], P[S][S/2], D[S][S/2], B[S][S/2];
    int m=-1, n=-1, i,j, colonneP = 0, colonneD = 0;

    while (m<0) {
        printf("\nImmetti un valore intero positivo minore di 100 per m\n");
        scanf("%d", &m);
    }
    while (n<0) {
        printf("\n\nImmetti un valore intero positivo minore di 100 per n\n");
        scanf("%d", &n);
        printf("\n\n");
    }
    for (i=0;i<m;i++)
        for (j=0;j<n;j++){
            printf("\nImmetti un valore intero per l'elemento %d,%d della matrice A: ", i, j);
            scanf("%d", &A[i][j]);
        }

    for (j=0;j<n;j=j+2) {
        for (i=0;i<m;i++)
            P[i][colonneP]=A[i][j];
        colonneP++;
    }
    for (j=1;j<n;j=j+2) {
        for (i=0;i<m;i++)
            D[i][colonneD] = A[i][j];
        colonneD++;
    }
}
```

```

}

if (colonneP == colonneD)
    for (i=0;i<m;i++)
        for (j=0;j<colonneP;j++)
            B[i][j] = P[i][j]-D[i][j];

/* codice non necessario, serve per verificare
che il programma si comporti correttamente */
for (i=0;i<m;i++) {
    for (j=0;j<colonneP;j++)
        printf("%d ", P[i][j]);
    printf("\n");
}
printf("\n\n");
for (i=0;i<m;i++) {
    for (j=0;j<colonneD;j++)
        printf("%d ", D[i][j]);
    printf("\n");
}
printf("\n\n");
if (colonneP == colonneD)
    for (i=0;i<m;i++) {
        for (j=0;j<colonneP;j++)
            printf("%d ", B[i][j]);
        printf("\n");
    }
}

```

### Esercizio 3 (4 punti)

Si considerino i tre frammenti di codice:

<pre>float a, prod=1; scanf("%f",&amp;a); while (a&gt;0) {     prod=prod*a;     scanf("%f",&amp;a); }</pre>	<pre>float a, prod=1; scanf("%f",&amp;a); while (a&gt;0) {     prod=prod*a;     a=a-1; }</pre>	<pre>float a, prod=1; a = 1; while (a&gt;0){     scanf("%f",&amp;a);     if (a&gt;0)         prod=a*prod; }</pre>
<b>A</b>	<b>B</b>	<b>C</b>

Due frammenti di programma sono da considerarsi equivalenti se producono lo stesso risultato sui valori delle variabili coinvolte.

Rispondere alle seguenti domande giustificando ciascuna risposta (risposte prive di un'adeguata giustificazione non verranno prese in considerazione):

1. I frammenti A e B sono equivalenti?
2. I frammenti A e C sono equivalenti?
3. I frammenti B e C sono equivalenti?

#### **Soluzione**

A e B non sono equivalenti fra di loro, mentre A è equivalente a C e, di conseguenza, B non è equivalente a C.

Sia A che C chiedono l'inserimento di una sequenza di valori float che termina non appena l'utente inserisce un valore negativo o uguale a zero. In A il primo valore viene letto all'esterno del ciclo, mentre in C viene letto all'interno del ciclo. In entrambi i frammenti, comunque, la variabile **prod** contiene alla fine il prodotto di tutti i valori inseriti da tastiera **escluso** l'ultimo valore float inserito.

Invece, in B viene richiesto l'inserimento di un solo valore float **a** da tastiera e, se questo è positivo, il frammento calcola il prodotto dei valori compresi tra **a** e 0.

#### Esercizio 4 (4 punti)

1. Si fornisca la codifica binaria in complemento a 2 del numero 191 utilizzando il minor numero di bit necessari per una corretta rappresentazione
2. Si fornisca la codifica binaria in virgola mobile secondo lo standard IEEE 754-1985 a precisione singola del numero 191.015625
3. Si dica, giustificando la risposta, se la rappresentazione fornita al punto 2 è esatta oppure comporta qualche approssimazione

#### Soluzione

1.  $191 : 2 = 1 \Rightarrow \underline{0}10111111$  (9 bit, aggiungo uno zero all'inizio perché il numero è positivo)

95	1
47	1
23	1
11	1
5	1
2	0
1	1
0	

2.  $191 : 2 = 1 \Rightarrow 10111111$

95	1
47	1
23	1
11	1
5	1
2	0
1	1
0	

$0,015625 \times 2 = 0$	$\Rightarrow$	$0,000001$
0,031250	0	
0,06250	0	
0,1250	0	
0,250	0	
0,5	0	
0	1	

$$191 = 10111111$$

$$0,015625 = 0,000001$$

Sommando:

$$191,015625 = 10111111,000001 = 1,0111111000001 \times 2^7$$

$$S = 0 \text{ (1 bit)}$$

$$E = e + K = 7 + 127 = 134 = 1000110 \text{ (8 bit)}$$

$$M = 01111110000010000000000 \text{ (23 bit)}$$

$$01000011001111110000010000000000 \text{ (32 bit)}$$

3. La rappresentazione fornita al punto 2 non comporta approssimazioni, né per periodicità, né per limitazioni nel numero di bit, ed è quindi esatta.