

| | | | | | | | |
|---|---|---------|--|--|--|--|--|
|  | Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 16 Novembre 2011 | | COGNOME E NOME | | | | |
| | RIGA | COLONNA | MATRICOLA | | | | |
| | | | <i>Spazio riservato ai docenti</i> | | | | |
| | | | <table border="1"> <tr> <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 ricalcare 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
 - **Peso Medio**: 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
 - **Numero CuccioliGenerati**: un intero
 - **Numero di esemplari presenti nel parco**: un intero
2. Si dichiari il tipo **archivioMamm**, in grado di rappresentare l'insieme dei mammiferi presenti nel parco (si ipotizzi che ci siano al massimo 50 tipi diversi di mammiferi).
3. Si dichiari la variabile **mammiferi** di tipo **archivioMamm**
4. Si scriva in linguaggio C, aggiungendo eventualmente opportune dichiarazioni di variabili, il frammento di codice che costruisce l'archivio degli esemplari di elefante sterili. In particolare, un elefante si dice sterile se è nato prima del 1990 e non ha concepito nessun cucciolo.

Soluzione

1.

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

```
typedef struct{
    int codiceEsemplare;
    int annoNascita;
    int numCuccioli;
} Esemplare;
```

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

2.

```
typedef Mammifero archivioMamm[30]
```

3.

```
archivioMamm mammiferi;
```

4.

```
// Variabili aggiuntive
int numMammiferi; // Numero di elementi nell'array archMammiferi.
Mammifero archMammiferiSterili[30];
int numMammiferiSterili = 0;
```

```
for (i = 0; i < numMammiferi; i++)
    if(strcmp(mammiferi[i].nome, "elefante") == 0)
        for(j=0; j<mammiferi[i].numEsemplari; j++)
            {
                if(mammiferi[i].esemplari[j].annoNascita < 1990 && mammiferi[i].esemplari[j].
                    numCuccioli == 0)
                    {
                        archMammiferiSterili[numMammiferiSterili] = mammiferi[i];
                        numMammiferiSterili++;
                    }
            }
}
```

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, $n \times m$ valori interi che memorizza in una matrice \mathbf{A} di dimensioni massime 100×100
- Costruisce una matrice \mathbf{P} di valori interi contenente solo le righe con indice pari della matrice \mathbf{A} e una matrice \mathbf{D} di valori interi contenente solo le righe 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

$m = 4$ $n = 3$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ -7 & -8 & -9 \\ 10 & 11 & 12 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 2 & 3 \\ -7 & -8 & -9 \end{bmatrix} \quad D = \begin{bmatrix} 4 & 5 & 6 \\ 10 & 11 & 12 \end{bmatrix} \quad B = \begin{bmatrix} 5 & 7 & 9 \\ 3 & 3 & 3 \end{bmatrix}$$

Soluzione

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

void main() {
    int A[S][S], P[S/2][S], D[S/2][S], B[S/2][S];
    int m=-1, n=-1, i, j, righeP=0, righeD=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 (i=0;i<m;i=i+2){
    for (j=0;j<n;j++)
        P[righeP][j]=A[i][j];
    righeP++;
}
for (i=1;i<m;i=i+2){
    for (j=0;j<n;j++)
        D[righeD][j]=A[i][j];
    righeD++;
}

if (righeP==righeD) {
    for (i=0;i<righeP;i++)
        for (j=0;j<n;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<righeP;i++) {
    for (j=0;j<n;j++)
        printf("%d ", P[i][j]);
    printf("\n");
}
printf("\n\n");
for (i=0;i<righeD;i++) {
    for (j=0;j<n;j++)
        printf("%d ", D[i][j]);
    printf("\n");
}
printf("\n\n");
if (righeP == righeD)
    for (i=0;i<righeP;i++) {
        for (j=0;j<n;j++)
            printf("%d ", B[i][j]);
        printf("\n");
    }
}

```

Esercizio 3 (4 punti)

Si considerino i tre frammenti di codice:

| | | |
|--|---|---|
| <pre>int x, sum=0; scanf("%d",&x); while (x>0) { sum=sum+x; scanf("%d",&x); }</pre> | <pre>int x, sum=0; x = 1; while (x>0) { scanf("%d",&x); sum=sum+x; }</pre> | <pre>int x, sum=0; scanf("%d",&x); while (x>0) { sum=sum+x; x=x-1; }</pre> |
| A | B | C |

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

Tutti e tre i programmi hanno un diverso comportamento e quindi non sono equivalenti fra di loro.

Sia A che B chiedono l'inserimento di una sequenza di valori interi che termina non appena l'utente inserisce un valore intero negativo o uguale a zero. Tuttavia, in A la variabile **sum** alla fine del frammento contiene la somma di tutti i valori inseriti da tastiera **escluso** l'ultimo valore inserito. Invece, in B la variabile **sum** contiene la somma di tutti i valori inseriti da tastiera **incluso** l'ultimo valore inserito. Infine, in C viene richiesto l'inserimento di un solo valore intero **x** da tastiera e se questo è positivo, il frammento calcola la somma dei primi **x** numeri interi.

Esercizio 4 (4 punti)

1. Si fornisca la codifica binaria in complemento a 2 del numero -221 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 -221.0625
3. Si dica, giustificando la risposta, se la rappresentazione fornita al punto 2 è esatta oppure comporta qualche approssimazione

Soluzione

1. $221 : 2 = 1 \Rightarrow 11011101 \Rightarrow \underline{0}11011101 \Rightarrow$ invertendo e sommando 1, $\Rightarrow 100100011$ (9 bit)

```
110 0
 55 1
 27 1
 13 1
  6 0
  3 1
  1 1
  0
```

2. $221 : 2 = 1 \Rightarrow 11011101$

```
110 0
 55 1
 27 1
 13 1
  6 0
  3 1
  1 1
  0
```

$0,0625 \times 2 = 0 \Rightarrow 0,0001$

```
0,1250    0
0,250     0
0,50      1
0
```

$221 = 11011101$

$0,0625 = 0,0001$

Sommando:

$221,0625 = 11011101,0001 = 1,10111010001 \times 2^7$

$S = 1$ (1 bit)

$E = e + K = 7 + 127 = 134 = 10000110$ (8 bit)

$M = (1,)10111010001(+ 12 zeri)$ (23 bit)

$11000011010111010001(+ 12 zeri)(32 \text{ bit})$

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