

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 20 Novembre 2013		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA

Spazio riservato ai docenti

--	--	--	--

- Il presente plico contiene 3 esercizi e deve essere debitamente compilato con cognome e nome, numero di matricola e posizione durante lo scritto (comunicata dal docente).
- Il tempo a disposizione è di 75 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) assicurandosi 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 implementi in C un programma che svolga le seguenti operazioni:

- chieda all'utente di inserire due valori interi R e C (entrambi maggiori di 0 e non più grandi di 100);
- chieda all'utente di popolare una matrice M di dimensioni massime 100x100 con RxC interi positivi letti da tastiera;
- stampi a video il minor numero di elementi dispari contenuti in una riga della matrice M.

Si aggiungano tutte le dichiarazioni di costanti, tipi e variabili necessarie per risolvere l'esercizio.

R,C e ciascun elemento della matrice M devono essere letti da tastiera; nel caso di inserimento di un valore non ammissibile, deve essere chiesto all'utente di immettere nuovamente il valore.

Esempio:

Dati i seguenti valori di R,C e M

$$R = 4 \quad C = 3 \quad M = \begin{matrix} & 2 & 4 & 1 \\ & 3 & 5 & 6 \\ & 8 & 7 & 10 \\ & 12 & 14 & 9 \end{matrix}$$

si osserva che la prima, la terza e la quarta riga di M contengono un solo elemento dispari; la seconda riga contiene invece 2 elementi dispari. Quindi il programma stampa a video: "Il minor numero di elementi dispari in una riga di M è 1".

Nota: Nell'esempio viene mostrata la sola porzione di M che contiene dati significativi.

Soluzione

```
#include <stdio.h>
#define MAX_DIM 100
typedef int matr[MAX_DIM][MAX_DIM];
typedef int array[MAX_DIM];
void main()
{
    matr M;
    array A; /* A[i], con 0<=i<MAX_DIM, contiene il numero di elementi dispari presenti
               nella riga i-esima di M */
    int i, j, min, R, C;
    /* Acquisizione di R t.c. R>0 e R<=100 */
    do
    {
        printf("Inserisci il numero di righe: ");
        scanf("%d", &R);
    } while(R<=0 || R>100);
    /* Acquisizione di C t.c. C>0 e C<=100 */
    do
    {
        printf("Inserisci il numero di colonne: ");
        scanf("%d", &C);
    } while(C<=0 || C>100);

    /*inizializzazione del vettore A*/
    for (i=0; i<MAX_DIM; i++) {
        A[i]=0;
    }

    /* Acquisizione dei valori della matrice M e riempimento di A con il numero di
       elementi dispari contenuti in ogni riga di M */
    for(i=0; i<R; i++)
        for(j=0; j<C; j++)
        {
            do
            {
                printf("Inserisci l'elemento di posizione [%d, %d]: ", i, j);
                scanf("%d", &M[i][j]);
            } while(M[i][j]<0);
```

```

        if (M[i][j]%2!=0)
        {
            A[i]++;
        }
    }
    /*calcolo del minimo in A*/
    min = A[0];
    for (i=1; i<R; i++) {
        if (A[i]<min) {
            min = A[i];
        }
    }
    printf("Il minor numero di elementi dispari in una riga di M e': %d ",min);
}

```

Esercizio 2 (6 punti)

Si definiscano i tipi di dato necessari a rappresentare in C il contatto di una rubrica. Ogni contatto deve includere le seguenti informazioni:

- Nome
- Cognome
- Cellulare
- Età
- Mail

Dichiarare due variabili chiamate *rubrica* e *selezione*. Tali variabili sono di tipo array di contatti di dimensione MAX (dove MAX è una costante definita opportunamente).

Ipotizzando che la variabile *rubrica* sia stata in precedenza riempita con i dati di MAX contatti, si scriva un frammento di codice C che:

- legga da tastiera un cognome;
- cerchi in *rubrica* tutti i contatti con quel cognome e li copi, senza lasciare buchi, in *selezione*;
- stampi a video l'età media dei contatti copiati in *selezione*.

Nota1: nella soluzione non è richiesto di includere il frammento di codice per riempire la variabile rubrica.

Nota2: per il confronto tra cognomi, si utilizzi la funzione strcmp(<stringa1>, <stringa2>) che restituisce 0 se le due stringhe sono uguali.

Soluzione

```
#define MAX 100
#define MAX_STRINGA 40

typedef char stringa[MAX_STRINGA];
typedef struct
{
    stringa nome;
    stringa cognome;
    stringa cellulare;
    int eta;
    stringa mail;
} contatto;

typedef contatto arrayContatti[MAX];

void main()
{
    arrayContatti rubrica, selezione;
    int i, k;
    stringa c;
    float etaMedia;

    /* la parte non richiesta relativa al riempimento di rubrica va qui ...*/

    /* lettura del cognome */
    scanf("%s", c);

    /* Inizializzazione dell'indice associato a selezione e di etaMedia */
    k = 0;
    etaMedia = 0;

    /* ricerca in rubrica, riempimento di selezione e calcolo dell'età media */
    for(i=0; i<MAX; i++)
        if(strcmp(c, rubrica[i].cognome) == 0)
        {
            selezione[k] = rubrica[i];
            etaMedia = etaMedia+selezione[k].eta;
            k++;
        }
    etaMedia = etaMedia/k;
    printf("età media delle persone con cognome %s: %d\n", c, etaMedia);
}
```

Esercizio 3 (4 punti)

- A. Si determini la codifica del valore -16.15 secondo lo Standard IEEE 754-1985 a precisione singola, riportando i calcoli effettuati.
- B. La codifica di -16.15, calcolata al punto A, è esatta? Giustificare la risposta

Soluzione

16/2	0
8/2	0
4/2	0
2/2	0
1/2	1

$(16)d = (10000)b$

0.15*2	0.3
0.3*2	0.6
0.6*2	1.2
0.2*2	0.4
0.4*2	0.8
0.8*2	1.6
0.6*2	1.2

rip

$(0.15)d = (00\underline{1001})b$

$(16.15)d = (10000.00\underline{1001})b = 1.000000\underline{1001} * 2^4$

L'esponente $E = k + 4 = 127 + 4 = 131$

131/2	1
65/2	1
32/2	0
16/2	0
8/2	0
4/2	0
2/2	0
1/2	1

0

$(131)d = (10000011)b$

In definitiva:

$S = 1$

$E = 10000011$

$M = 00000010011001100110011$

La codifica è approssimata poiché la parte di mantissa viene troncata (i 23 bit non possono rappresentare completamente il valore di mantissa che ha una parte periodica che si ripete all'infinito)