

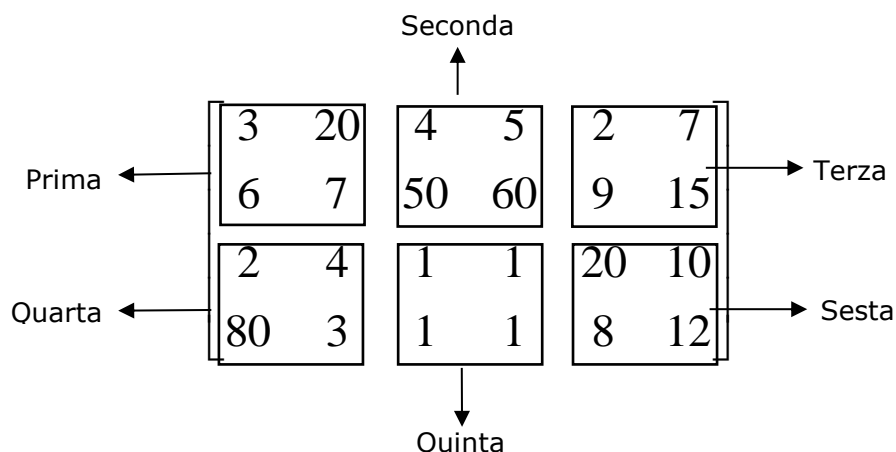
|  |  |         |  |  |  |  |  |
|--|--|---------|--|--|--|--|--|
|  | Politecnico di Milano<br>Facoltà di Ingegneria Industriale<br><b>INFORMATICA B</b><br>Appello 30 Giugno 2014 |         | COGNOME E NOME   |  |  |  |  |
|  | RIGA   | COLONNA | MATRICOLA  |  |  |  |  |
|  |  |         | Spazio riservato ai docenti<br><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 3 esercizi e **deve essere debitamente compilato con cognome e nome, numero di matricola.**
- Il tempo a disposizione è di 1 ora e mezza.
- 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.**
- Per accedere all'orale è necessario impostare sufficientemente il primo ed il secondo esercizio e raggiungere almeno 14 punti nello scritto. L'orale è parte integrante dell'esame e consente di ottenere fino a un massimo di sei punti. Il punteggio dell'orale si aggiunge a quello dello scritto per comporre il voto finale.

## Esercizio 1 (10 punti)

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

- Acquisisca da tastiera due valori interi  $nRighe$  e  $nCol$ . I due valori devono essere pari, maggiori di 0 e minori di  $N$ , dove  $N$  è una costante che deve essere opportunamente definita con valore 100. L'acquisizione deve essere ripetuta finché l'utente non inserisca valori corretti.
- Acquisisca da tastiera  $nRighe \times nCol$  valori interi e li memorizzi in una matrice  $m$ , di dimensioni massime  $N \times N$ , organizzandoli su un numero di righe pari a  $nRighe$  e un numero di colonne pari a  $nCol$ .
- Considerando la matrice  $m$  come suddivisa in sottomatrici  $2 \times 2$ , calcoli e stampi a video il numero di sottomatrici  $2 \times 2$  con media maggiore di 10. Per esempio, si consideri il caso della matrice  $m$  in figura:



Questa matrice ha 4 righe e 6 colonne, e contiene quindi le 6 sottomatrici evidenziate nella figura. La prima sottomatrice ha media 9, la seconda 29.75, la terza 8.25, la quarta 22.25, la quinta 1 e la sesta 12.5. Il valore che deve essere stampato a video per tale matrice  $m$  è quindi 3.

## Soluzione

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

```
int main(){
```

```
    int m[N][N], nRighe, nCol, i, j, cont = 0;
    float media;
```

```
    /* Acquisizione dei numeri di righe e colonne */
```

```
    do{
        printf("Inserire il numero di righe: ");
        scanf("%d", &nRighe);
    } while(nRighe <= 0 || nRighe >= N || (nRighe % 2 != 0));
```

```
    do{
        printf("Inserire il numero di colonne: ");
        scanf("%d", &nCol);
    } while(nCol <= 0 || nCol >= N || (nCol % 2 != 0));
```

```
    /* Acquisizione della matrice */
```

```
    for(i = 0; i < nRighe; i++){
        for(j = 0; j < nCol; j++){
            printf("Inserire l'elemento in posizione (%d, %d): ", i, j);
            scanf("%d", &m[i][j]);
        }
    }
```

```
}  
  
/* Conteggio delle sottomatrici con media maggiore di 10 */  
for(i = 0; i < nRighe; i = i + 2){  
    for(j = 0; j < nCol; j = j + 2){  
        media = (m[i][j] + m[i+1][j] + m[i][j+1] + m[i+1][j+1]) / 4.0;  
        if(media > 10)  
            cont++;  
    }  
}  
  
printf("Ci sono %d sottomatrici con media maggiore di 10.", cont);  
  
return 0;  
}
```

## Esercizio 2 (10 punti)

Si consideri l'array definito in Matlab e chiamato *dati*. Tale array contiene le informazioni riguardanti le precipitazioni atmosferiche registrate da una stazione meteorologica. In particolare, ogni elemento dell'array *dati* è una struct che contiene un campo numerico *giorno*, un campo numerico *mese*, un campo numerico *anno*, e un campo numerico *mm* che rappresenta la quantità di pioggia (in mm) caduta nella data specificata dai precedenti campi. Ad esempio, per memorizzare nell'array *dati* che il 3 giugno 2013 sono stati registrati 15 mm di pioggia si avrà:

```
dati(i) = struct("giorno", 3, "mese", 6, "anno", 2013, "mm", 15);
```

Si risponda ai seguenti quesiti:

- Scrivere l'intestazione di una funzione Matlab *media* che riceva in ingresso l'array *dati* e un parametro numerico *anno*; la funzione deve restituire in uscita un array di 12 elementi che, sulla base del contenuto dell'array *dati*, fornisca il valore medio dei mm di pioggia in ciascun mese dell'anno solare *anno*.
- Scrivere una funzione Matlab di ordine superiore *mostra* che riceva in ingresso l'array *dati*, un parametro funzione *f* e un parametro numerico *anno*. Si assuma che *f* abbia la stessa intestazione della funzione *media* al punto A.  
La funzione *mostra* deve applicare *f* all'array *dati* per calcolare una statistica mese per mese dell'anno solare specificato dal parametro *anno*; inoltre, la funzione *mostra* deve visualizzare i valori di tale statistica mensile mediante un grafico, con il mese in ascissa e il valore della statistica per ciascun mese in ordinata.
- Scrivere un esempio di chiamata della funzione *mostra* implementata al punto precedente, con cui si vuole mostrare il grafico delle precipitazioni medie mensili dell'anno 2012, i cui valori sono contenuti in un array *dati* che deve essere caricato con i valori memorizzati in un file in formato Matlab di nome "*dati*".
- Scrivere il corpo della funzione Matlab *media* assumendo che il suo parametro contenga per ogni mese degli anni considerati almeno un dato relativo alle precipitazioni.

## Soluzione

```
A. function stat = media (dati,anno)
```

```
B. function mostra (dati,f,anno)
```

```
    stat = f(dati,anno);
```

```
    plot (1:12,stat);
```

```
C. load 'dati' dati;
```

```
    mostra(dati,@media,2012);
```

```
D. function stat = media (dati,anno)
```

```
    for i = 1:12
```

```
        idx = [dati.anno] == anno & [dati.mese] == i;
```

```
        stat(i) = mean([dati(idx).mm]);
```

```
    end
```

### Esercizio 3 (6 punti)

1. Si determini la codifica del valore 8.0 secondo lo Standard IEEE 754-1985 a precisione singola, riportando i calcoli effettuati.
2. Si determini la codifica del valore 0.4 secondo lo stesso standard, sempre riportando i calcoli effettuati.
3. Si consideri il seguente programma in linguaggio C e si indichi l'effetto della sua esecuzione, motivando adeguatamente la risposta.

```
#include <stdio.h>

int main() {
    float f;
    int i;

    f = 0.4;

    for (i = 1; i < 20; i++)
        f = f + 0.4;

    printf("\n\nIl numero 0.4 * 20 ");
    if (f != 8.0) printf("non ");
    printf("e' uguale a %f\n\n", 8.0);

    return 0;
}
```

### Soluzione

1.  
 $8_{10} = 1000_2$ , e inoltre la parte frazionaria del numero è nulla, quindi la forma normalizzata è  $1.000000000000000000000000 \cdot 2^3$  e la mantissa è composta da 23 zeri, come sempre avviene per i numeri che sono uguali a una potenza di 2.  
La caratteristica, in eccesso K, con  $K = 127$ , è  $c = 3_{10} + 127_{10} = 130_{10} = 10000010_2$ . Quindi la rappresentazione di 8 è  $0\ 10000010\ 000000000000000000000000$ . Si noti che la rappresentazione è esatta, nel senso che non sono state introdotte approssimazioni.

2.  
Per il numero 0.4 si ottiene la rappresentazione della parte frazionaria con i seguenti calcoli

$$\begin{array}{r} 0.4 \ X \\ \hline 2 \\ \hline 0 \leftarrow 0.8 \\ 0.8 \ X \\ \hline 2 \\ \hline 1 \leftarrow 1.6 \\ 0.6 \ X \\ \hline 2 \\ \hline 1 \leftarrow 1.2 \\ 0.2 \ X \\ \hline 2 \\ \hline 0 \leftarrow 0.4 \end{array}$$

Poiché si è riottenuto il numero 0.4 di partenza, la rappresentazione in base 2 è periodica,  $0.4_{10} = 0.0\underline{110}_2$  (il periodo è mostrato sottolineato) e la rappresentazione in forma normalizzata è  $1.\underline{1001}_2 \cdot 2^{-2}$ . La mantissa è perciò  $10011001100110011001100$ , la caratteristica è  $c = -2_{10} + 127_{10} = 125_{10} = 01111101_2$ . Quindi la rappresentazione di 0.4 è  $0\ 01111101\ 10011001100110011001100$ . Si noti che la rappresentazione è approssimata, per via del fatto che la parte frazionaria è periodica.

3.

Viene visualizzata la scritta

`Il numero 0.4 * 20 non e' uguale a 8.0`

a causa degli errori che si accumulano durante le operazioni di somma, poiché l'addendo è rappresentato in modo approssimato, e la somma viene confrontata col numero 8.0 che invece è rappresentato in modo esatto.