

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Appello 1 settembre 2015		COGNOME E NOME				
	RIGA	COLONNA	MATRICOLA				
			Spazio riservato ai docenti <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.**
- L'esame orale è parte integrante dell'esame e deve essere realizzato almeno sufficientemente per il superamento dell'esame complessivo.

Esercizio 1 (10 punti)

In una gara di tuffi, N giudici esprimono il loro giudizio su K tuffi (con $N \leq 5$ e $K \leq 8$). Il giudizio è un valore numerico intero tra 0 e 10. La valutazione finale assegnata a ogni tuffo è dato dalla somma dei giudizi espressi dai giudici su quel tuffo.

Si scriva un programma in linguaggio C che, inserendo da tastiera i valori di N e K e il giudizio assegnato da ogni giudice a ogni tuffo, determini il tuffo migliore (cioè quello con valutazione finale più alta) e il giudice più severo (cioè quello che ha assegnato mediamente i voti più bassi ai tuffi). Il programma poi mostri a video il tuffo migliore con la sua valutazione finale e il giudice più severo con la media dei suoi giudizi (in caso di parità, si consideri uno qualsiasi dei tuffi migliori o dei giudici più severi).

Soluzione

```
#include <stdio.h>
#define MAXN 5
#define MAXK 8

int main()
{
    int voti[MAXK][MAXN], tuffo[MAXK], giudice[MAXN];
    int K, N, i, j, posmin = 0, posmax = 0;

    // Carico i valori di N e K
    do {
        printf("Inserire K (valori tra 1 e 8, compresi): ");
        scanf("%d", &K);
    } while(K <= 0 || K > 8);

    do {
        printf("Inserire N (valori tra 1 e 5, compresi): ");
        scanf("%d", &N);
    } while (N <= 0 || N > 5);

    // Carico i giudizi assegnati da ogni giudice a ogni tuffo
    for (i = 0; i < K; i++)
        for (j = 0; j < N; j++)
            scanf("%d", &voti[i][j]);

    // Inizializzo i valori dei tuffi e le medie dei giudizi dei giudici
    for (i = 0; i < K; i++)
        tuffo[i] = 0;

    for (j = 0; j < N; j++)
        giudice[j] = 0;

    // Calcolo la somma dei valori di ogni tuffo e la somma dei giudizi di ogni giudice
    for (i = 0; i < K; i++) {
        for (j = 0; j < N; j++)
        {
            tuffo[i] += voti[i][j];
            giudice[j] += voti[i][j];
        }
    }

    // Cerco il tuffo migliore e il giudice più severo e li stampo con i loro valori
    for (i = 1; i < K; i++)
        if (tuffo[i] > tuffo[posmax])
            posmax = i;

    for (j = 1; j < N; j++)
        if (giudice[j] < giudice[posmin])
            posmin = j;

    printf("Il tuffo %d ha il valore piu' alto, pari a %d\n",posmax+1, tuffo[posmax]);
    printf("Il giudice %d e' il piu' severo (media %f)\n",posmin+1,
            giudice[posmin]/((float)K));

    return 0;
}
```

Esercizio 2 (10 punti)

Punto A

Normalizzare un vettore v consiste nel trasformarne gli elementi del vettore applicando la seguente operazione:

$$v_i = (v_i - \min_v) / (\max_v - \min_v),$$

dove v_i è l' i -esimo elemento di v , \min_v e \max_v sono rispettivamente il minimo e il massimo valore di v .

Evitando l'utilizzo di cicli e sfruttando le proprietà di Matlab, implementare una funzione Matlab *normUnif* che riceve in ingresso un vettore numerico v e lo restituisce normalizzato dopo aver applicato la trasformazione qui sopra.

Soluzione A

```
function w = normUnif(v)
    minv = min(v);
    maxv = max(v);
    w = (v - minv) / (maxv - minv);
```

Punto B

Un altro modo di normalizzare un vettore v è applicare la seguente trasformazione:

$$v_i = \frac{v_i - m_v}{\sqrt{s_v}},$$

dove $m_v = \sum_1^n v_i / n$, $s_v = (\sum_1^n (v_i - m_v)^2) / (n - 1)$ ed n è la dimensione del vettore v .

Evitando l'utilizzo di cicli e sfruttando le proprietà di Matlab, implementare una funzione Matlab *normGauss* che riceve in ingresso un vettore numerico v e lo restituisce normalizzato dopo aver applicato la trasformazione qui sopra. Nell'implementazione non è consentito utilizzare le funzioni Matlab *mean*, *var* e *stdev*.

Soluzione B

```
function w = normGauss(v)
    n = length(v);
    mv = sum(v) / n;
    sv = sum((v - mv).^2) / (n-1);
    w = (v - mv) / sqrt(sv);
```

Punto C

Evitando l'utilizzo di cicli e sfruttando le proprietà di Matlab, implementare una funzione Matlab di ordine superiore *plotNorm* che riceve in ingresso un vettore v e una variabile f che contiene l'handle di una funzione che esegue la normalizzazione di un vettore (come ad esempio le funzioni *normUnif* e *normGauss* implementate in precedenza). La funzione *plotNorm* dovrà prima di tutto applicare la funzione f al vettore v ; poi dovrà rappresentare nel piano cartesiano un insieme di punti le cui ascisse siano gli elementi originali di v e le cui ordinate siano gli elementi di v dopo aver applicato la normalizzazione.

Soluzione C

```
function plotNorm(v, f)
    w = f(v);
    plot(v, w);
```

Esercizio 3 (6 punti)

Un calcolatore è dotato di un sistema di memoria centrale e memoria cache con le seguenti caratteristiche:

- Miss Rate = 25 %
- Miss Penalty = 800 ns
- Hit Time = 40 ns

Rispondere alle seguenti domande (giustificando i risultati ottenuti con gli opportuni calcoli):

- Calcolare il tempo medio di accesso ai dati del sistema con memoria cache.
- Mantenendo invariati Miss Rate e Hit Time, qual è il valore massimo che può avere Miss Penalty affinché il tempo medio sia inferiore a 200 ns?

Soluzione

$$a) TM = (1 - 0.25) * 40 + 0.25 * 800 = 30 + 200 = 230 \text{ ns}$$

$$b) (1 - 0.25) * 40 + 0.25 * x < 200 \\ 30 + 0.25 * x < 200 \rightarrow x < 170/0.25 \rightarrow x < 680 \text{ ns}$$

Affinchè la cache offra un vantaggio di prestazioni, il Miss Penalty deve essere inferiore a 680 ns