

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 5 febbraio 2015		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA
			Spazio riservato ai docenti <div style="border: 1px solid black; width: 100px; height: 20px; margin-left: auto;"></div>

- Il presente plico contiene 3 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 1 ora e 45 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!) **facendo in modo 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)

In un teatro di Milano la temperatura nella sala è regolata in modo automatico; un sensore monitora la temperatura registrando un valore ogni minuto. I valori letti in una giornata vengono troncati a due cifre decimali, inseriti in un array **temp** e salvati in un file dati di Matlab chiamato "**log.mat**".

Un esempio di valori contenuti nel file "**log.mat**" è il seguente:

```
temp = [22.00 22.50 23.20 21.45 22.00 22.35 23.00 23.40];
```

La caldaia si accende solo quando la temperatura rilevata è strettamente inferiore a una certa soglia e si spegne non appena viene raggiunta tale soglia.

1. Scrivere in Matlab una funzione **calcolaCosto** che:
 - riceve in ingresso un vettore di temperature **temperature**, un valore soglia **soglia** e un parametro "**costoAlMinuto**" che indica il costo al minuto del gas consumato dalla caldaia;
 - restituisca il costo totale "**costoTotale**" del gas consumato e un vettore **minutiAccesa** contenente gli indici dell'array **temp** corrispondenti ai minuti nei quali la caldaia è rimasta accesa.

Per il calcolo del costo totale del gas consumato si consideri che se il totale dei minuti nei quali la caldaia ha funzionato supera i 30 minuti (non necessariamente consecutivi), allora il **costoAlMinuto** va diminuito del 20%.

Ad esempio, nel caso la funzione **calcolaCosto** riceva in ingresso il valori di temperatura contenuti nel file "**log.mat**" dell'esempio sopra riportato, una soglia di temperatura pari a 23.00 e un valore di **costoAlMinuto** pari a 100, ritornerà un **costoTotale** di 500 e il seguente vettore **minutiAccesa** = [1 2 4 5 6].

2. Scrivere poi uno script che:
 - legga da tastiera il valore della variabile **soglia**;
 - legga dal file "**log.mat**" i valori di temperatura contenuti nell'array **temp**;
 - legga da tastiera il valore della variabile **costoAlMinuto**;
 - crei un vettore "**x**" che che indichi i minuti in cui sono state effettuate le misurazioni;
 - disegni sullo stesso grafico (con titolo del grafico e dei due assi) la temperatura al variare del tempo, evidenziando sul grafico stesso:
 - o il valore soglia con una retta orizzontale nera a doppio spessore;
 - o i minuti nei quali la caldaia ha funzionato, marcandoli con il simbolo 'X' rosso (senza congiungerli).
 - stampi a video il costo totale del gas consumato e i minuti nei quali la caldaia è rimasta accesa.

Soluzione

1.

```
function [costoTotale minutiAccesa] = calcolaCosto(temp, soglia, costoAlMinuto)
```

```
minutiAccesa = find(temp < soglia);
```

```
if (length(minutiAccesa) > 30)
    costoTotale = length(minutiAccesa) * costoAlMinuto * 0.80;
else
    costoTotale = length(minutiAccesa) * costoAlMinuto;
end
```

2.

```
clear
close all
```

```

clc

load log.mat temp;
soglia = input('Inserire la soglia: ');
costoAlMinuto = input('Inserire il costo al minuto: ');

x = 1:length(temp);

figure()
hold on
ylabel('temperatura');
xlabel('minuto');
title('Temperatura sala');
plot(x, temp);

y = soglia * ones(size(x));
plot(x, y, 'k', 'LineWidth', 2);

[costoTotale, minutiAcceso] = calcolaCosto(temp, soglia, costoAlMinuto);

temp1 = temp(minutiAcceso);
x1 = x(minutiAcceso);
plot(x1, temp1, 'Xr');

disp(['Costo totale del gas consumato: ', num2str(costoTotale), ' Euro.']);
disp('La caldaia ha funzionato nei seguenti minuti: ');
disp(minutiAcceso);

```

Esercizio 2 (6 punti)

Si considerino i seguenti esempi di matrici costruite da “quadrati concentrici”:

matr1 =

2	2	2	2	2	2
2	3	3	3	3	2
2	3	4	4	3	2
2	3	4	4	3	2
2	3	3	3	3	2
2	2	2	2	2	2

matr2 =

2	2	2	2	2
2	5	5	5	2
2	5	1	5	2
2	5	5	5	2
2	2	2	2	2

Come si vede dagli esempi, si tratta di matrici quadrate aventi valori costanti in cornici concentriche. Si sviluppi in MATLAB una funzione ricorsiva **corniciConcentriche** che, data una generica matrice, restituisca 1 se la matrice è costituita da cornici concentriche, 0 altrimenti.

Per esempi, in matrice *matr1*:

- la prima cornice concentrica è composta solo da elementi col valore 2
- la seconda cornice concentrica è composta solo da elementi col valore 3
- la terza cornice concentrica è composta solo da elementi col valore 4

Per sviluppare questa funzione si assuma di avere a disposizione la funzione **valoriDiCorniceUguale** che, data una matrice quadrata, restituisce 1 se tutti i valori disposti sulla sua cornice più esterna (costituita dalla prima e dall'ultima riga e dalla prima e dall'ultima colonna) sono uguali tra loro, 0 altrimenti.

Per esempio: **valoriDiCorniceUguale(matr1)** restituisce 1.

NB: non si chiede di sviluppare **valoriDiCorniceUguale**. Ci si focalizzi solo sulla funzione ricorsiva.

Soluzione

```
function [ris] = corniciConcentriche(m)

[r, c] = size(m);
if r == 1 || r == 0
    ris = 1;
elseif valoriDiCorniceUguale(m)
    ris = corniciConcentriche(m(2:end-1, 2:end-1));
else
    ris = 0;
end

%NB: l'implementazione di valoriDiCorniceUguale viene fornita per
%completezza ma non è richiesta dall'esercizio

function [ris] = valoriDiCorniceUguale(m)

[r, s] = size(m);
if r == 0
    ris = 0;
else
    v = [m(1, :) m(:, 1)' m(end, :) m(:, end)'];
    if all(v == v(1))
        ris = 1;
    end
end
```

```
        else
            ris = 0;
        end
    end
end
```

Esercizio 3 (4 punti)

Si considerino due dispositivi aventi le seguenti configurazioni

Configurazione A

- 1 Mbyte di memoria fisica e pagine di memoria da 4 Kbyte
- una memoria cache con hit rate di 0.8, hit time di 50 ns e miss penalty di 150 ns

Configurazione B

- indirizzo di memoria fisica a 24 bit e pagine di memoria da 64 Kbyte
- una memoria cache con hit rate di 0.9, hit time di 40 ns e miss penalty di 250 ns

Domande:

- 1) Quali dei due dispositivi ha il maggior numero di pagine di memoria fisica?
- 2) Si può dire quale dei due dispositivi ha maggiore memoria fisica? E quale dei due avrà maggiore memoria virtuale?
- 3) In quale dei due dispositivi l'accesso alla memoria è più rapido?
- 4) Si consideri ora il dispositivo più lento: qual è l'hit rate minimo che dovrebbe avere per essere rapido almeno quanto l'altro?

Giustificare tutte le risposte riportando gli opportuni calcoli.

Soluzione

1) Nessuno dei due dispositivi; entrambi hanno lo stesso numero di pagine di memoria fisica (2^8):

- La configurazione A ha 20 bit di memoria fisica indirizzabile ($1 \text{ Mbyte} = 2^{20}$), di cui 12 dedicati all'offset ($2^{12} = 4 \text{ K}$), quindi A ha 2^8 pagine di memoria fisica.
- La configurazione B ha 24 bit di memoria fisica indirizzabile, di cui 16 dedicati all'offset all'interno della pagina ($2^{16} = 64 \text{ K}$), quindi ha 2^8 pagine di memoria fisica.

2) La configurazione B ha più memoria fisica perché $2^{24} = 16 \text{ Mbyte} > 1 \text{ Mbyte}$.

Non è possibile invece dire nulla della memoria virtuale perché le informazioni fornite non ci permettono di risalire al numero di pagine virtuali né della configurazione A né di quella B.

3) La configurazione B è più veloce:

- La configurazione A ha tempo medio di accesso alla memoria di $0.8 * 50 \text{ ns} + 0.2 * 150 \text{ ns} = 70 \text{ ns}$
- La configurazione B ha tempo medio di accesso alla memoria di $0.9 * 40 \text{ ns} + 0.1 * 250 \text{ ns} = 61 \text{ ns}$

4) Se hit rate = x, dovrà essere $x * 50 \text{ ns} + (1 - x) * 150 \text{ ns} \leq 61 \text{ ns}$

$$- 100 * x \text{ ns} \leq - 89 \text{ ns}$$

$$x \geq 0.89 \text{ quindi hit rate minimo} = 0.89.$$