

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Appello del 6 Febbraio 2012		COGNOME E NOME				
	RIGA	COLONNA	MATRICOLA				
			<i>Spazio riservato ai docenti</i>				
			<table border="1" style="display: inline-table;"> <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 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 2h.
- 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 l'ammissione all'esame orale è necessario aver almeno impostato sufficientemente **entrambi** gli esercizi 1 e 2.
- L'esame orale è parte integrante dell'esame e deve essere realizzato almeno sufficientemente per il superamento dell'esame complessivo.

Esercizio 1 (10 punti)

Sia **M** una matrice di dimensione $N \times N$ (N costante simbolica pari a 3) di numeri interi appartenenti all'intervallo $[0,9]$.

1. Si scriva un frammento di programma in linguaggio C (e le relative dichiarazioni di variabili) in grado di individuare il valore intero **num** che compare più frequentemente nella matrice **M** (si ipotizzi che tale valore **num** sia sempre unico) e di visualizzare a schermo il contenuto della matrice **M**, mostrando però un trattino ("-") al posto dei valori interi diversi da **num**. Ad esempio, se la matrice **M** fosse inizializzata come segue:

```
0 1 2
3 1 4
1 0 1
```

a video dovrebbe essere visualizzato:

```
- 1 -
- 1 -
1 - 1
```

2. Utilizzando la sottostante definizione di tipo **coord** e la variabile **Posizioni** di tipo array di $N \times N$ **coord**, si scriva il frammento di programma C che salva in elementi consecutivi dell'array **Posizioni**, a partire dal primo elemento, le coordinate (riga, colonna) di tutte le celle della matrice **M** in cui è presente il valore **num**.

```
typedef struct{
    int riga;
    int colonna;
}coord;

coord Posizioni[N*N];
```

SOLUZIONE PUNTO 1

```
#define RANGE 10
#define N 3
...
int M[N][N];
int freq[RANGE];
int i, j, k;
int max, valmax;

for(i = 0; i < N*N; i++)
    freq[i]=0;
for(i = 0; i < N; i++)
    for(j = 0; j < N; j++)
        freq[M[i][j]]++;

max = 0;
for(k = 0; k < RANGE; k++)
    if(freq[k] > max){
        max = freq[k];
        valmax = k;
    }

printf("\n Valore %d con frequenza massima %d\n", valmax, max);

for(i = 0; i < N; i++)
    for(j = 0; j < N; j++) {
        if(M[i][j] == valmax)
            printf("%d", M[i][j]);
```

```
else
    printf("_");
if(j < N-1)
    printf("\t");
else
    printf("\n");
}
```

SOLUZIONE PUNTO 2

```
k = 0;
for(i = 0; i < N; i++)
    for(j = 0; j < N; j++)
        if(M[i][j] == valmax) {
            Posizioni[k].riga = i;
            Posizioni[k].colonna = j;
            k++;
        }
```

Esercizio 2 (10 punti)

- a) Si scriva in linguaggio MATLAB una funzione `controllaMultipli` che prende in ingresso una matrice **M** di numeri interi strettamente maggiori di zero e un numero intero positivo **n**, e restituisce al programma chiamante un vettore colonna **V** avente lo stesso numero di righe della matrice **M**. La funzione determina, per ogni riga della matrice **M**, qual è il massimo numero di valori multipli di **n** che compaiono consecutivamente in colonne adiacenti della riga della matrice **M** e assegna tale numero alla posizione corrispondente in **V**. Di conseguenza, l'*i*-esimo elemento del vettore **V** conterrà il massimo numero di valori multipli di **n** consecutivi nell'*i*-esima riga della matrice **M**.

Per esempio

$$\mathbf{M} = \begin{array}{|c|c|c|c|c|} \hline 12 & 19 & 4 & 21 & 6 \\ \hline 9 & 18 & 6 & 5 & 18 \\ \hline 3 & 11 & 9 & 3 & 9 \\ \hline 11 & 12 & 5 & 6 & 4 \\ \hline 13 & 1 & 4 & 23 & 1 \\ \hline 7 & 12 & 4 & 9 & 18 \\ \hline \end{array} \quad \text{e } n = 3; \quad \text{la funzione restituisce} \quad \mathbf{V} = \begin{array}{|c|} \hline 2 \\ \hline 3 \\ \hline 3 \\ \hline 1 \\ \hline 0 \\ \hline 2 \\ \hline \end{array}$$

perchè nella prima riga individua come multipli di 3 tre valori (12, 21, 6), di cui due (21 e 6) sono consecutivi, nella seconda riga individua come multipli di 3 quattro valori (9, 18, 6, 18), di cui tre (9, 18, 6) sono consecutivi, e così via fino all'ultima riga.

Per determinare se un valore è multiplo di un altro si può utilizzare la funzione `mod` che restituisce il resto della divisione intera tra due numeri (per esempio, `mod(5, 2)` restituisce 1).

- b) Si scriva un frammento di script che crea una matrice con i valori dell'esempio e chiama la funzione `controllaMultipli` passando come parametro la matrice creata e il valore 3. Si faccia in modo che il risultato ottenuto per effetto della chiamata alla funzione venga stampato a schermo.

Soluzione punto a

```
function V = controllaMultipli(M, n)

m = mod(M, n);
[nRows, nCols] = size(M);
V = zeros(nRows, 1);
for ii = 1: nRows
    counter = 0;
    for jj = 1 : nCols
        if (m(ii, jj) == 0)
            counter = counter + 1;
        else
            counter = 0;
        end
        V(ii) = max([V(ii), counter]);
    end
end
```

Soluzione punto b

```
matrice = [12 19 4 21 6; 9 18 6 5 18; 3 11 9 3 9; 11 12 5 6 4; 13 1 4 23 1; 7 12 4 9 18];
controllaMultipli(matrice, 3)
```


Esercizio 3 (8 punti)

Si consideri la seguente funzione ricorsiva in linguaggio Matlab:

```
function [ris] = mistero(v, n)
    if (n > 1)
        v2 = v(mod(v, n) ~= 0 | v == n);
        ris = mistero(v2, n-1);
    else
        ris = v;
    end
```

Rispondere ai seguenti quesiti giustificando opportunamente le risposte.

1. Dire qual è il contenuto della variabile `x` dopo aver eseguito la chiamata:
`x = mistero([2 3 4 5 7 9 11 13 15], 10)`
2. Implementare un frammento di programma in linguaggio Matlab che legga da tastiera un numero intero positivo `A`. Quindi, chiami la funzione `mistero` passandole, come primo argomento, un vettore contenente tutti i numeri interi compresi fra 2 e `A` e, come secondo argomento, il numero `A`. Infine, stampi a video il valore ritornato dalla chiamata alla funzione `mistero` precedentemente descritta.
3. Spiegare in maniera sintetica quale problema risolve il frammento di programma implementato al punto 2.

Soluzione

1. `x = [2 3 5 7 11 13]`
2. `A = input('Inserisci un numero intero positivo: ');`
`ris = mistero(2:A, A);`
`disp(ris);`
3. Il frammento di programma stampa a video tutti i numeri primi compresi fra 2 e un numero intero positivo `A` inserito dall'utente

Esercizio 4 (6 punti)

Si ha un sistema con le seguenti caratteristiche:

- 4 processori
- 16 GB di RAM il cui tempo medio di accesso totale è di 350 ns
- Gestione della memoria con indirizzo virtuale di 128 bit e con pagine da 32 KB
- Una memoria cache con Hit Rate 60%, Hit Time 20 ns, Miss Penalty 900ns.

Si risponda alle seguenti domande giustificando adeguatamente tutte le risposte:

1. È possibile che in questo sistema esistano tre processi in stato di esecuzione?
2. Qual è la struttura degli indirizzi della memoria fisica e virtuale? Si indichino in modo preciso i campi che li compongono e le loro dimensioni.
3. La memoria cache fornita è in grado di migliorare in modo significativo le prestazioni del sistema?

Soluzione

1. Sì, può accadere ad esempio quando tre processi hanno il controllo di tre dei 4 processori del sistema.

2. Per indirizzare 16 GB (=234 byte) di memoria RAM fisica servono 34 bit di indirizzo fisico. Per indirizzare le pagine da 32 KB (= 215 byte) servono 15 bit di offset. Gli indirizzi virtuali sono da 128 bit, per cui:

MEMORIA FISICA (bit indirizzo totale: 34 bit):

$NPF = 34 \text{ bit} - 15 \text{ bit} = 19 \text{ bit}$

offset = 15 bit

MEMORIA VIRTUALE (bit indirizzo totale: 128 bit):

$NPV = 128 \text{ bit} - 15 \text{ bit} = 113 \text{ bit}$

offset = 15 bit

3. No, perché il tempo di accesso alla memoria in presenza della cache è:

$TACC = HR * HT + (1-HR) * MP = 0.6 * 20 \text{ ns} + 0.4 * 900 \text{ ns} = 12 \text{ ns} + 360 \text{ ns} = 372 \text{ ns}$

che è maggiore di quello della memoria RAM (350 ns). In questo caso la memoria cache aumenta i tempi di accesso alla memoria ed è perciò non adeguata al sistema dato.