

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 3 Febbraio 2009		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA

- Il presente plico contiene 4 esercizi, deve essere debitamente compilato con cognome e nome, numero di matricola, posizione durante lo scritto (comunicata dal docente).
- Il tempo a disposizione è di 2 ore.
- 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, purché in modo ben marcato e 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.
- Gli studenti degli anni precedenti (e solo loro) *possono* svolgere per gli esercizi 1 e 2 le versioni in C.

Esercizio 1 (6 punti)

Scrivere una funzione che prende come parametro due matrici A e B, della stessa dimensione, e produce i seguenti 3 valori:

- 1) Il numero di elementi uguali in posizioni corrispondenti.
- 2) Il numero di elementi di A *minimisti* per la matrice B, cioè il numero di posizioni, nella matrice A, in cui è presente un elemento che è minore o uguale ad almeno un elemento presente nella matrice B.
- 3) il numero massimo di elementi uguali in posizioni corrispondenti che sono presenti in una stessa colonna

Per esempio, se A e B sono definite come segue:

A = [1 2 3; 3 4 5; 1 2 3; 6 3 7];

B = [4 1 3; 3 4 5; 0 1 2; 6 3 9];

Il numero di elementi uguali in posizioni corrispondenti è 6: l'elemento di posizione (1, 3); tutti gli elementi della seconda riga; gli elementi di posizione (4, 1) e (4, 2).

Il numero di elementi di A minimisti per B è pari a 12, infatti tutti gli elementi di A sono minori di almeno un elemento di B.

Il numero massimo di elementi uguali in una colonna è pari a 2, infatti in ogni colonna ci sono due elementi in A e B che sono uguali e si trovano nella stessa posizione. Se per esempio, modificassimo la matrice B in modo tale che l'elemento di posizione (3, 3) sia pari a 3, avremmo che in terza colonna ci sono tre elementi in A e B che sono uguali e si trovano nella stessa posizione. Di conseguenza, Il numero massimo di elementi uguali in una colonna diventerebbe pari a 3.

Per sviluppare la funzione è possibile utilizzare le funzioni descritte nella Tabella 1. È inoltre possibile utilizzare altre funzioni di libreria disponibili in Matlab/Octave o funzioni definite al momento, purché venga spiegato chiaramente quale è lo scopo della funzione usata, che cosa riceve come parametri di ingresso e cosa produce in uscita.

Nota per gli studenti degli anni precedenti: Chi ha seguito negli anni precedenti può svolgere questo esercizio in C (ed appoggiarsi ad altre funzioni C già sviluppate o da sviluppare ma delle quali si spiegherà lo scopo e l'intestazione). Per capire l'esempio si tenga presente che in Matlab/Octave gli indici di riga e colonna delle matrici partono dal valore 1 e che nell'assegnamento di valore alle matrici ciascun elemento è separato dagli altri tramite uno spazio (o una virgola) e le righe sono separate da ";".

Tabella 1

<p>max() <i>Sintassi</i> max(A) <i>Descrizione</i> Se A è un vettore, max(A) restituisce l'elemento più grande in A Se A è una matrice, max(A) restituisce un vettore riga che contiene gli elementi più grandi di ciascuna colonna della matrice A <i>Esempio</i> max([1,5;3,4;5,2]) restituisce [5,5]</p>	<p>min() <i>Sintassi</i> min(A) <i>Descrizione</i> Se A è un vettore, min(A) restituisce l'elemento più piccolo in A Se A è una matrice, min(A) restituisce un vettore riga che contiene gli elementi più piccoli di ciascuna colonna della matrice A <i>Esempio</i> min([1,5;3,4;5,2]) restituisce [1,2]</p>
<p>any() <i>Sintassi</i> any(A) <i>Descrizione</i> Se A è un vettore, any(A) restituisce 1 (true) se A contiene almeno un valore numerico diverso da 0 o un 1 (true), altrimenti restituisce 0 (false) Se A è una matrice, any(A) restituisce un vettore riga logico in cui ciascun elemento è un 1 (true) se la corrispondente colonna di A contenga almeno un valore numerico diverso da 0 o un 1 (true). <i>Esempio</i> any([0,1,0;0,0,1]) restituisce [0,1,1]</p>	<p>all() <i>Sintassi</i> all(A) <i>Descrizione</i> Se A è un vettore, all(A) restituisce 1 (true) se A contiene solo valori numerici diversi da 0 o solo 1 (true), altrimenti restituisce 0 (false) Se A è una matrice, all(A) restituisce un vettore riga logico in cui ciascun elemento è un 1 (true) se la corrispondente colonna di A contenga solo valori numerici diversi da 0 o solo 1 (true). <i>Esempio</i> all([0,1,0;0,1,1]) restituisce [0,1,0]</p>
<p>sum() <i>Sintassi</i> sum(A) <i>Descrizione</i> Se A è un vettore, sum(A) restituisce la somma degli elementi in A Se A è una matrice, sum(A) restituisce un vettore riga contenente la somma degli elementi di ciascuna colonna di A <i>Esempio</i> sum([1,5,7;3,4,2]) restituisce [4,9,9]</p>	<p>size() <i>Sintassi</i> size(A) <i>Descrizione</i> Se A è un vettore, size(A) restituisce la lunghezza del vettore Se A è una matrice, size(A) restituisce un vettore riga contenente il numero di righe e di colonne della matrice A <i>Esempio</i> size([1,5,7;3,4,2]) restituisce [2,3]</p>

Soluzione

Versione Matlab/Octave

Nota: riportiamo una soluzione compatta. Altre soluzioni meno compatte ma corrette sono ritenute accettabili.

Osserviamo che un elemento della matrice A è minore o uguale a qualche elemento della matrice B se e solo se è minore o uguale al massimo fra di essi. Quindi per verificare che un elemento di A sia minimista per B è sufficiente confrontarlo con l'elemento massimo di B, calcolato preventivamente e *una tantum*.

```
function [r1,r2,r3]=soluzioneEs1b(A, B)
t1 = A==B;
M = max(max(B));
t2 = A <= M;
```

```
r1 = sum(sum(t1));
r2 = sum(sum(t2));
r3 = max(sum(t1));
```

Versione C

```
#define N 20
```

```
typedef enum {false, true} boolean;
```

```

/* minimista restituisce vero se il valore x e` minore di qualche valore in matr. nrighe ed ncolonne
indicano il numero di righe e colonne di matr che sono occupate con valori significativi */
boolean minimista(int x, int matr[][N], int nrighe, int ncolonne);

void cm1(int A[][N], int B[][N], int nrighe, int ncolonne, int *r1, int *r2, int *r3)
{
    int i, j, ugualiCol;
    int uguali = 0, minimisti = 0, maxUguali = 0;
    for (j=0; j<ncolonne; j++)
    {
        ugualiCol = 0;
        for (i=0; i<nrighe; i++)
        {
            if(A[i][j]==B[i][j])
                ugualiCol = ugualiCol + 1;
            if(minimista(A[i][j], B, nrighe, ncolonne)==true)
                minimisti = minimisti+1;
        }
        if(ugualiCol > maxUguali) maxUguali = ugualiCol;
        uguali = uguali + ugualiCol;
    }
    /* preparazione dei valori che vengono prodotti per il chiamante attraverso il passaggio di
    parametri per indirizzo */
    *r1 = uguali;
    *r2 = minimisti;
    *r3 = maxUguali;
}

```

Esercizio 2 (3 punti)

Facendo uso della funzione di ordine superiore acc spiegata a lezione (e di cui si riporta il codice qui sotto) codificare la funzione modulo(v) che prende come argomento un vettore di numeri $v=[v_1, v_2, \dots, v_n]$ e

calcola e restituisce come risultato il valore $\sum_{i=1}^n v_i / \prod_{i=1}^n v_i$

Codice della funzione acc.

```

function [x]=acc(f, a, u)
    x = u;
    for i=1:length(a)
        x = f(x, a(i));
    end

```

Soluzione

```

function ris = modulo (v)
    prod = @(x, y) x*y;
    sum = @(x, y) x+y;
    ris = acc(sum, v, 0)/acc(prod, v, 1);

```

Esercizio 2 solo per gli studenti degli anni precedenti (3 punti)

Si consideri una lista dinamica di interi strettamente positivi, i cui elementi sono di tipo ELEMENTO definito di seguito:

```

typedef struct El {
    int dato;
    struct El *next;
} ELEMENTO;
typedef ELEMENTO *lista;

```

Si codifichi in C una funzione prodotto avente il seguente prototipo:

```
int prodotto(lista Testa, int M)
```

Tale funzione riceve come parametro la testa della lista e un intero M e restituisce il prodotto dei soli valori della lista che sono divisibili per M.

Soluzione

```
/* versione ricorsiva */
int prodotto(lista Testa, int M)
{
    if(Testa == NULL) return 1;
    else if(Testa->dato%M == 0)
        return Testa->dato*prodotto(Testa->next, M);
    else return prodotto(Testa->next, M);
}
```

```
/* versione iterativa */
int prodotto(lista Testa, int M)
{
    int p=1;
    while(Testa!=NULL)
    {
        if(Testa->dato%M == 0)
            p = p*Testa->dato;
        Testa = Testa->next;
    }
    return p;
}
```

Esercizio 3 (4.5 punti)

Si considerino i due seguenti numeri X e Y, codificati secondo lo standard virgola mobile IEEE visto a lezione.

```
X      s = 0
        m = 000000000000000000000000
        c = 10000000
Y      s = 0
        m = 00110011001100110011001
        c = 01010101
```

Si calcoli, nel modo più semplice possibile (cioè facendo un minimo di calcoli aritmetici) la rappresentazione del loro quoziente Y / X

Soluzione

Poiché X ha mantissa composta da soli zeri, il suo valore è una potenza di 2; la sua caratteristica vale 128 (1 seguito da sette zeri vale $2^7=128$) quindi codifica l'esponente di valore 1 (=128-127); perciò il valore di X è esattamente 2 (per via del bit di segno 0); in conclusione $Y/X = Y/2$, e la rappresentazione di questo valore ha la stessa mantissa di X, e l'esponente diminuito di 1. Questo significa che il bit meno significativo della caratteristica viene modificato da 1 a 0.

```
Y/2    s = 0
        m = 00110011001100110011001
        c = 01010100
```

Esercizio 4 (3.5 punti)

Un sistema dispone di 16 Kbyte di memoria fisica indirizzabile; inoltre è dotato di memoria virtuale con paginazione caratterizzata dai seguenti parametri: l'indirizzo virtuale è di 16 bit e le pagine sono di 512 byte.

Rispondere alle seguenti domande giustificando le risposte:

- qual è la dimensione della memoria virtuale indirizzabile?
- definire la struttura dell'indirizzo logico e di quello fisico indicando la lunghezza dei campi che li costituiscono

Soluzione

- a) la memoria virtuale indirizzabile è pari a 64 Kbyte (2^{16} byte)
- b) Indirizzo Virtuale: 16 bit NPV: 7 bit offset virtuale: 9 bit
Indirizzo Fisico: 14 bit NPF: 5 bit offset fisico: 9 bit