	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 23 Gennaio 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 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** (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 (6 punti)

1. Implementare una funzione $r = \text{cerchio}(x, y, rg)$. La funzione riceve in ingresso due vettori x e y di uguale dimensione che rappresentano le ascisse e le ordinate di un insieme di punti nel piano cartesiano e un valore rg che rappresenta il raggio di una circonferenza centrata sull'origine del piano cartesiano; la funzione $\text{cerchio}(x, y, rg)$ restituisce un vettore logico r , di lunghezza pari quella dei vettori x e y , tale che $r(i)$ è pari a 1 se e solo se il punto definito dalle coordinate $x(i)$ e $y(i)$ si trova all'esterno della circonferenza di raggio rg e centro $(0,0)$.

Suggerimento. Un punto si trova all'esterno di una circonferenza se la sua distanza dal centro è superiore al raggio della circonferenza.

2. Si consideri una famiglia di funzioni in linguaggio Matlab che, ricevendo come parametro un numero intero $N > 0$, restituiscano due vettori x e y , di lunghezza N , contenenti rispettivamente le ascisse e le ordinate di N punti nel piano cartesiano. Un esempio di funzione di questo tipo è il seguente:

```
[x,y] = puntiSuAsseX(N)
for k = 1 : 1 : N
    x(k) = k;
    y(k) = 0;
end;
```

La chiamata `puntiSuAsseX(5)` restituisce i due vettori:

```
X = [1 2 3 4 5] y = [0 0 0 0 0]
```

Implementare la funzione di ordine superiore $M = \text{conta}(N, \text{genera}, rg)$ che riceve in ingresso: nel parametro N un valore numerico intero positivo non nullo, nel parametro genera una funzione, per esempio la `puntiSuAsseX`, e nel parametro rg un secondo valore numerico intero positivo non nullo.

La funzione conta utilizza la funzione nel parametro genera per produrre N punti nel piano cartesiano e, utilizzando la funzione `cerchio` sviluppata al punto 1, restituisce, tramite il parametro M , il **numero dei punti che si trovano all'esterno della circonferenza di raggio rg e centro $(0,0)$** . Per esempio, la chiamata `conta(5, @puntiSuAsseX, 2)` restituisce il valore 3, corrispondente ai punti $(3,0)$, $(4,0)$ e $(5,0)$ prodotti dalla chiamata di `puntiSuAsseX(5)` che si trovano all'esterno della circonferenza data.

Soluzione

Punto1.

```
function r = cerchio(x, y, rg)
    r = sqrt(x.*x + (y).*(y)) > rg;
```

Punto 2.

```
function M = conta(N, genera, rg)
    [x,y] = genera(N);
    M = sum(cerchio(x, y, rg));
```

Esercizio 2 (5 punti)

In algebra lineare la **matrice compagna** del polinomio di grado n

$$P(X) = c_0 + c_1 X + \dots + c_{n-1} X^{n-1} + X^n$$

è la matrice quadrata $n \times n$ avente:

- 1 sulla prima sottodiagonale
- i coefficienti di P , cambiati di segno, sull'ultima colonna
- e 0 altrove:

$$C_P^t = \begin{pmatrix} 0 & 0 & \dots & 0 & -c_0 \\ 1 & 0 & \dots & 0 & -c_1 \\ 0 & 1 & \dots & 0 & -c_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -c_{n-1} \end{pmatrix}$$

Scrivere una funzione che, ricevuto in ingresso il vettore dei coefficienti $[c_0 \ c_1 \ \dots \ c_{n-2} \ c_{n-1}]$ di P , restituisce la corrispondente matrice compagna.

Soluzione

```
function CP = matriceCompagna(coefficients)
n = size(coefficients,2);
CP = zeros(n);
CP(2:n+1:end) = 1;
CP(1:n,n) = -coefficients;
```

Esercizio 3 (3 punti)

Un sistema dispone di 256 Kbyte di memoria virtuale indirizzabile con paginazione, caratterizzata dai seguenti parametri: indirizzo fisico di 16 bit e pagine di dimensione di 2 Kbyte.

Rispondere alle seguenti domande giustificando le risposte:

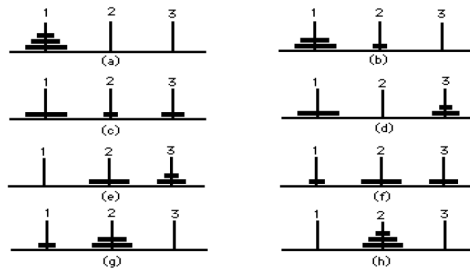
- a) Quale è la dimensione della memoria fisica indirizzabile?
- b) Quale è la struttura dell'indirizzo virtuale e di quello fisico, e la lunghezza dei campi che li costituiscono?

Soluzione

- a) La memoria fisica indirizzabile è pari a 64 Kbyte (2^{16} byte)
- b) Indirizzo Virtuale: 18 bit NPV: 7 bit offset: 11 bit
Indirizzo Fisico: 16 bit NPF: 5 bit offset: 11 bit

Esercizio 4 (3 punti)

Il gioco delle Torri di Hanoi utilizza una tavoletta con tre pioli e un certo numero di dischetti di diametro diverso infilati sullo stesso piolo in ordine di diametro decrescente, col dischetto più piccolo posto in cima. Obiettivo del gioco è ricomporre la stessa configurazione di dischetti su un piolo diverso da quello iniziale, spostando un disco alla volta da un piolo all'altro, con la restrizione che un dischetto non può mai essere appoggiato su di un altro dischetto di diametro inferiore. Mostriamo in figura la soluzione per il caso di tre dischetti da spostare dal primo al secondo piolo, rappresentata attraverso la sequenza delle successive configurazioni di dischetti sui diversi pioli, indicate con le lettere alfabetiche progressive dalla (a) alla (h).



Nella soluzione ricorsiva del problema, lo spostamento di una pila di n dischetti (con $n > 1$) da un piolo a un altro, usando il terzo piolo come supporto intermedio, segue lo schema seguente:

1. Si sposta sul piolo intermedio la pila di $n-1$ dischetti.
2. Poi si sposta il dischetto più grande sul piolo di destinazione, lasciando libero il piolo di partenza.
3. Infine si sposta sul piolo di destinazione la pila di $n-1$ dischetti che si trova sul piolo intermedio.

Nel caso in cui $n-1 > 1$, lo spostamento della pila di $n-1$ dischetti sul piolo intermedio e poi dal piolo intermedio a quello finale richiede l'esecuzione di una nuova chiamata ricorsiva.

Il caso di una pila di n dischetti con $n = 1$ viene risolto banalmente spostando l'unico dischetto dal piolo di partenza a quello di destinazione.

La seguente funzione ricorsiva:

```
function [] = hanoi(n, da, a, per)
%primo parametro numero dei dischi
%altri parametri pioli di partenza, di arrivo e intermedio
if (n > 1) hanoi(n-1, da, per, a); end;
fprintf('Sposta un disco dal piolo %d al piolo %d \n', da, a);
if (n > 1) hanoi(n-1, per, a, da); end;
```

permette di stampare le istruzioni per risolvere il gioco delle Torri di Hanoi: per esempio, la chiamata `hanoi(3, 1, 2, 3)` produce la stampa seguente

```
Sposta un disco dal piolo 1 al piolo 2
Sposta un disco dal piolo 1 al piolo 3
Sposta un disco dal piolo 2 al piolo 3
Sposta un disco dal piolo 1 al piolo 2
Sposta un disco dal piolo 3 al piolo 1
Sposta un disco dal piolo 3 al piolo 2
Sposta un disco dal piolo 1 al piolo 2
```

Si vuole codificare una versione della funzione `hanoi` che, invece di visualizzare mediante l'istruzione `fprintf` le mosse, restituisca in uscita, tramite un opportuno parametro, una stringa che contiene la sequenza di tutte le mosse da effettuare (es. '1->2'), separate dalla stringa ' poi '. A tale scopo si vuole definire una versione della funzione `hanoi` con la seguente testata:

```
function [msg] = hanoiMsgPOI(n, da, a, per)
```

tale che, per esempio, la chiamata `hanoiMsgPOI(3, 1, 2, 3)` restituisca in uscita la seguente stringa:

```
1->2 poi 1->3 poi 2->3 poi 1->2 poi 3->1 poi 3->2 poi 1->2
```

Codificare in completo dettaglio la funzione `hanoiMsgPOI(n, da, a, per)`.

Soluzione

```
function [msg] = hanoiMsgPOI(n, da, a, per)
if (n > 1) msg1 = [hanoiMsgPOI(n-1, da, per, a) ' poi '];
else msg1 = [];
end;
msg2 = [num2str(da) '->' num2str(a) ];
if (n > 1) msg3 = [' poi ' hanoiMsgPOI(n-1, per, a, da)];
else msg3 = [];
end;
msg = [msg1 msg2 msg3];
```

