

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 6 Febbraio 2014		COGNOME E NOME				
	RIGA	COLONNA	MATRICOLA				
			<i>Spazio riservato ai docenti</i>				
			<table border="1"> <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 e 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) assicurandosi 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)

Per calcolare il valore della rata mensile di un mutuo si usa la seguente formula:

$$Rata = Capitale * tassoMens / (1 - (1 + tassoMens)^{-n})$$

Dove:

- Capitale = Capitale iniziale – capitale già pagato
- tassoMens = tasso d'interesse annuale / 12
- n = numero di rate ancora da pagare

- Si sviluppi la funzione Matlab/Octave **calcolaRata** che prende in ingresso il valore del capitale, del tasso d'interesse annuale (ad esempio, se il tasso d'interesse annuale è del 5% la funzione prende in ingresso il valore 0.05) e del numero di rate ancora da pagare, e restituisce al chiamante il valore della rata corrente.
- Si sviluppi uno script in Matlab/Octave che simula l'evoluzione della rata in base al variare del tasso d'interesse annuale e del capitale ancora da pagare. Lo script acquisisce dall'utente le seguenti informazioni:
 - Il valore del capitale iniziale del mutuo.
 - Il tasso iniziale d'interesse annuale.
 - Il numero di rate da pagare.

Lo script calcola il valore delle rate ancora da pagare simulando che vi sia, dopo ogni rata, un incremento dello 0.03% del tasso d'interesse annuale. Inoltre lo script presenta a schermo un diagramma che mostra l'andamento delle rate nel tempo. Infine lo script genera una tabella contenente i valori di tutte le rate e del decremento del capitale nel tempo e li salva su un unico file.

Soluzione

```
function [r] = calcolaRata(capitale, tassoAnnuale, n)
    if ~isscalar(capitale) || ~isreal(capitale)
        error('calcolaRata: InputCapitaleNonCorretto', 'Il capitale deve essere un valore reale non negativo.');
```

```
    end
    if ~isscalar(tassoAnnuale) || ~isreal(tassoAnnuale) || tassoAnnuale < 0
        error('calcolaRata: InputTassoAnnualeNonCorretto', 'Il tasso annuale deve essere uno scalare reale
                                                    non negativo.');
```

```
    end
    if ~isscalar(n) || ~isreal(n) || (floor(n) ~= n) || (n < 0)
        error('calcolaRata: InputNumeroRateNonCorretto', 'Il numero di rate deve essere un valore
                                                    intero non negativo.');
```

```
    end
    tassoMens = tassoAnnuale / 12;
    r = capitale * tassoMens / (1 - (1 + tassoMens)^-n);
```

```
%scriptMutuo
```

```
capitaleIniz = input('Inserisci il valore del capitale iniziale: ');
while (~isscalar(capitaleIniz) || capitaleIniz < 0 || ~isnumeric(capitaleIniz) ||
                                             fix(capitaleIniz) ~= capitaleIniz)
    disp('Il capitale iniziale deve essere uno scalare intero.');
```

```
    tassoAnnuale = input('Inserisci un altro valore per il capitale iniziale: ');
end
nRate = input('Inserisci il numero di rate mensili: ');
while (~isscalar(nRate) || nRate < 0 || ~isnumeric(nRate) || fix(nRate) ~= nRate)
    disp('Il numero di rate mensili deve essere uno scalare intero.');
```

```
    nRate = input('Inserisci un altro valore per il numero di rate mensili: ');
end
tassoAnnuale = input('Inserisci il tasso iniziale d'interesse annuale: ');
while (~isscalar(tassoAnnuale) || ~isreal(tassoAnnuale) || tassoAnnuale < 0)
    disp('Il tasso iniziale d'interesse annuale deve essere uno scalare reale non negativo.');
```

```
    tassoAnnuale = input('Inserisci un altro valore per il tasso iniziale d'interesse annuale: ');
end
capitale(1) = capitaleIniz;
rate(1) = calcolaRata(capitale(1), tassoAnnuale, nRate);
for x = 2:nRate
    tassoAnnuale = tassoAnnuale * 1.0003;
    capitale(x) = capitaleIniz - sum(rate(1:x-1));
    rate(x) = calcolaRata(capitale(x), tassoAnnuale, nRate - x + 1);
```

```
end
x = 1:nRate;
plot(x, rate, '*-');
tabella = [x' rate' capitale'];
save -ascii 'simulazioneMutuo' tabella;
```

Esercizio 2 (6 punti)

Un vettore x si definisce vettore triplo di un altro vettore y quando x e y sono di uguale lunghezza e ogni elemento di x è maggiore del triplo del corrispondente elemento di y .

Scrivere in Matlab/Octave una funzione **ricorsiva** `vettoreTriplo` che riceve in ingresso due vettori **colonna** a e b . La funzione `vettoreTriplo`:

- Restituisce -1 se a e b hanno lunghezza diversa, o se non sono due vettori colonna.
- Restituisce 0 se a e b sono due vettori colonna di uguale lunghezza, ma non è vero che ogni elemento di a è maggiore del triplo del corrispondente elemento di b .
- Restituisce 1 se a e b sono due vettori colonna di uguale lunghezza e ogni elemento di a è maggiore del triplo del corrispondente elemento di b .

Nota: Se x è vettore triplo di y , il vettore x_1 , costruito aggiungendo un elemento all'inizio del vettore x , è triplo del vettore y_1 se quest'ultimo è costruito aggiungendo all'inizio del vettore y un elemento che è minore di $1/3$ dell'elemento posto all'inizio del vettore x .

Esempi:

```
vettoreTriplo ([1; 2; 3; 4], [2 5 1 10]) restituisce -1
vettoreTriplo ([1 2 3 4; 5 3 5 5], [2; 2; 3; 9]) restituisce -1
vettoreTriplo ([7; 2; 4; 4], [2; 5; 1; 10]) restituisce 0
vettoreTriplo([7; 16; 40; 31], [2; 5; 13; 10]) restituisce 1
```

Nell'implementazione non è consentito utilizzare cicli, e quindi non si possono usare le istruzioni *while* e *for*. Inoltre, non è possibile utilizzare alcuna funzione di libreria disponibile in Matlab/Octave ad eccezione di ***length*** e ***size***.

Soluzione

```
function ris = vettoreTriplo(a, b)

if size(a, 2) ~= 1 || size(b, 2) ~= 1 || length(a) ~= length(b)
    ris = -1;
elseif a(1) <= 3 * b(1)
    ris = 0;
elseif length(a) == 1 && a(1) > 3 * b(1)
    ris = 1;
else
    ris = vettoreTriplo(a(2:end), b(2:end));
end
```

Esercizio 3 (4 punti)

Un sistema dispone di 64 Kbyte di memoria fisica e una memoria virtuale indirizzabile con paginazione, caratterizzata dai seguenti parametri: indirizzo virtuale di 16 bit e 128 pagine di memoria.

Rispondere alle seguenti domande giustificando le risposte:

- a) Quale è la dimensione della memoria virtuale indirizzabile?
- b) Quale è la struttura dell'indirizzo virtuale e di quello fisico, e la lunghezza dei campi che li costituiscono?
- c) Un consulente afferma che raddoppiando la dimensione delle pagine di memoria e quindi riducendo il numero di pagine virtuali, si riducono i possibili sprechi di memoria. Siete d'accordo con questa affermazione? Argomentare in maniera adeguata la propria risposta.

Soluzione

- a) 16 bit -> 2^{16} byte di memoria virtuale -> 64 Kbyte
- b) 128 (2^7) pagine virtuali -> 7 bit per NPV
64 kByte memoria fisica -> 16 bit indirizzo fisico
Quindi si ha,
NPV: 7 bit, offset: 9 bit
NPF: 7 bit, offset: 9 bit
- c) L'affermazione non è corretta: aumentando la dimensione delle pagine di memoria, e quindi riducendo il numero di pagine virtuali, si ha meno controllo sull'utilizzo della memoria, dovendo assegnare blocchi contigui (una pagina) più grandi a un unico processo, aumenta la probabilità di sprecare spazi durante il funzionamento del sistema.