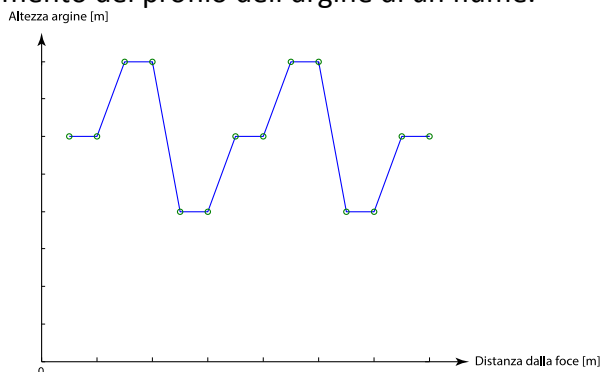
	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 28 gennaio 2013		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA
			Spazio riservato ai docenti <div style="border: 1px solid black; width: 100px; height: 20px; margin-left: auto;"></div>

- Il presente plico contiene 3 esercizi e **deve essere debitamente compilato con cognome e nome, numero di matricola.**
- Il tempo a disposizione è di 1 ora e 30.
- 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.**
- **Saranno valutate meno le soluzioni che NON sfruttano i vantaggi sintattici del linguaggio Matlab/Octave nella gestione degli array.**

Esercizio 1 (7 punti)

La seguente figura mostra l'andamento del profilo dell'argine di un fiume.



Come si nota, l'argine presenta altezze diverse nei vari punti del fiume. Si supponga che il fiume scorra in piano lungo tutto il suo percorso (non ci sono quindi salti di livello). Si vuole sviluppare in Matlab/Octave una funzione **calcolaRischio** che sia in grado di identificare e restituire al chiamante i punti dell'argine a rischio di esondazione in base a:

1. profilo dell'argine, rappresentato da un vettore *profiloArgine* che contiene le altezze dell'argine, in metri rispetto al fondo del letto del fiume, in vari punti del corso del fiume a distanze crescenti dalla sua foce (nella precedente figura i dati contenuti in *profiloArgine* sono cerchiati)
2. l'altezza attuale del fiume in metri rispetto al fondo del letto del fiume
3. i millimetri di incremento totale di altezza del fiume in base alla pioggia prevista per la giornata.

Nota 1: il vettore *profiloArgine* è tale che:

- *profiloArgine*(1) contiene l'altezza dell'argine a 1 metro dalla foce del fiume
- *profiloArgine*(2) contiene l'altezza dell'argine a 2 metri dalla foce del fiume
- *profiloArgine*(3) contiene l'altezza dell'argine a 3 metri dalla foce del fiume
- e così via.

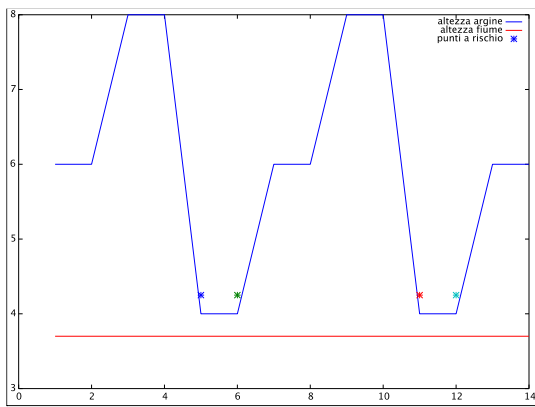
Nota 2: i punti dell'argine a rischio di esondazione sono quelli in cui il fiume potrebbe raggiungere un'altezza superiore a quella dell'argine. Li si indichi in termini della loro distanza dalla foce (in metri).

A) Si definisca l'intestazione della funzione **calcolaRischio**, spiegando il significato dei suoi parametri in ingresso e dei suoi valori di ritorno.

B) Si scriva il codice della funzione **calcolaRischio**.

C) Si costruisca uno script che:

- carica da un file ascii *profiloArgine.txt* (contenente un numero per ogni riga) i dati relativi alle altezze dell'argine del fiume nei vari punti;
- chiede all'utente le informazioni relative all'altezza attuale del fiume e alla previsione di incremento giornaliero totale di altezza del fiume in base alla pioggia prevista;
- richiama la funzione **calcolaRischio** sviluppata al punto precedente;
- visualizza sullo schermo un grafico che mostri: i) l'andamento del profilo dell'argine, ii) il livello attuale del fiume, iii) i punti dell'argine a rischio di esondazione, come rappresentato nella figura seguente:



- genera e visualizza a video una tabella contenente i soli punti dell'argine a rischio di esondazione; tale tabella deve contenere nella prima colonna la distanza del punto a rischio in metri dalla foce, nella seconda colonna l'altezza dell'argine in quel punto e, nella terza colonna, i minuti che passeranno tra il rilevamento e l'esondazione in quel punto. [Per calcolare quest'ultimo valore si usi la formula: $\text{minuti} = (\text{profiloArgine} - \text{altezzaFiume}) * 1000 / (\text{mmIncrementoPrevisto} / 24 * 60)$]

Soluzione

Funzione

```
function puntiARischio = calcolaRischio(profiloArgine, altezzaFiume, mmIncrementoPrevisto)
% function puntiARischio = calcolaRischio(profiloArgine, altezzaFiume, mmIncrementoPrevisto)
%
% input:
% profiloArgine:      vettore contenente l'altezza dei punti dell'argine
% altezzaFiume:      scalare contenente il valore dell'altezza attuale del fiume
% mmIncrementoPrevisto: scalare contenente il valore dell'incremento di altezza previsto
%                    del fiume
% output:
% puntiARischio:      vettore contenente i punti dell'argine a rischio di esondazione
%
% puntiARischio = find(profiloArgine <= altezzaFiume + (mmIncrementoPrevisto / 1000));
```

Script

```
load('profiloArgine.txt');
altezzaFiume = input('Altezza attuale del fiume in metri: ');
mmIncrementoPrevisto = input('Incremento totale di altezza del fiume in millimetri: ');
puntiARischio = calcolaRischio(profiloArgine, altezzaFiume, mmIncrementoPrevisto);
nCampioni = length(profiloArgine);
x = 1:nCampioni;
yAltezza = zeros(1, nCampioni);
yAltezza(:) = altezzaFiume;
plot(x, profiloArgine);
hold on
plot(x, yAltezza);
plot(puntiARischio, altezzaFiume+ mmIncrementoPrevisto /1000, '*');
hold off
metri = 1:length(profiloArgine);
minuti = 1000 * 24 * 60 * abs(profiloArgine - altezzaFiume) / (mmIncrementoPrevisto);
tabella = [metri', profiloArgine ,minuti];
tabella = tabella(puntiARischio, :)
disp(tabella);
```

Esercizio 2 (6 punti)

Punto 1) Scrivere la funzione ricorsiva **cifra** che riceve come parametri due numeri interi **num** e **k** (si supponga che entrambi i numeri siano sempre strettamente positivi). La funzione **cifra** restituisce la **k**-esima cifra del numero **num** a partire da destra. Per esempio:

- `cifra(1456, 1)` deve restituire 6
- `cifra(5136, 4)` deve restituire 5
- `cifra(512, 2)` deve restituire 1

Nota: Non è consentito l'uso della funzione `num2str` di Matlab/Octave.

Suggerimento: un numero intero positivo di n cifre, c_n, \dots, c_1 può essere rappresentato in termini di potenze di 10 come segue: $c_n \cdot 10^{n-1} + c_{n-1} \cdot 10^{n-2} + \dots + c_2 \cdot 10 + c_1$. Per esempio $1456 = 1 \cdot 10^3 + 4 \cdot 10^2 + 5 \cdot 10 + 6$. La cifra 1 del numero può essere ricavata calcolando il resto della divisione tra il numero e 10. Per esempio, il resto della divisione $1456/10$ è pari a 6.

Punto 2) Scrivere la funzione ricorsiva **cifraConControllo** che, in modo analogo alla funzione precedente, riceve come parametri due numeri interi **num** e **k** (si supponga che entrambi i numeri siano sempre strettamente positivi). La funzione **cifraConControllo** restituisce la **k**-esima cifra del numero **num** a partire da destra. Nel caso in cui **k** sia maggiore del numero effettivo di cifre che compongono **num**, la funzione restituisce -1. Per esempio:

- `cifraConControllo(512, 2)` deve restituire 1
- `cifraConControllo(123, 5)` deve restituire -1 (perché in questo caso **num** è composto da sole 3 cifre)

Nota: Non è consentito l'uso della funzione `num2str` di Matlab/Octave.

Soluzione

Punto 1)

```
function ris = cifra(num, k)

    if k == 1
        ris = mod(num, 10);
    else
        ris = cifra(floor(num/10), k-1);
    end
```

Punto 2)

```
function ris = cifraConControllo(num, k)

    if (k > 1 && num < 10)
        ris = -1
    else
        if k == 1
            ris = mod(num, 10);
        else
            ris = cifraConControllo(floor(num/10), k-1);
        end
    end
```

Esercizio 3 (4 punti)

Un sistema dotato solamente di memoria centrale ha un tempo di accesso ai dati pari a 300 ns. Per poter migliorare il tempo di accesso si decide di aggiungere una memoria cache dalle seguenti caratteristiche:

- Hit Rate = 60 %
- Hit Time = 50 ns
- Miss Penalty = 800 ns

Rispondere alle seguenti domande (giustificando i risultati ottenuti con gli opportuni calcoli):

- Calcolare il tempo medio di accesso ai dati.
- Qual è il valore minimo che deve avere l'Hit Rate affinché la cache offra un vantaggio in termini di prestazioni (cioè il tempo medio di accesso con la cache sia inferiore a 300 ns)?
- Il progettista suggerisce che è possibile incrementare l'Hit Rate con i seguenti accorgimenti:
 - in caso di miss è opportuno portare in cache non solo il dato richiesto, ma anche un (piccolo) blocco di dati che si trovano nelle posizioni immediatamente successive al dato richiesto
 - quando vengono caricati dei dati in cache, se la memoria cache è piena è opportuno che i nuovi dati sostituiscano quelli più recentemente caricati in cache.

Commentare gli accorgimenti (i) e (ii) proposti dal progettista e in particolare valutarne l'efficacia in riferimento ai principi utilizzati per gestire la memoria nei sistemi operativi.

Soluzione

a) $TM = 0.6 * 50 + 0.4 * 800 = 30 + 320 = 350 \text{ ns}$

b) $x * 50 + (1 - x) * 800 \leq 300$
 $750 * x \geq 500 \rightarrow x > (2 / 3) \sim 66 \% \text{ è l'Hit rate minimo}$

- c) L'accorgimento (i) è in accordo con il principio di località spaziale e può quindi portare a miglioramenti dell'hit rate. L'accorgimento (ii) invece è in disaccordo con il principio di località temporale e quindi potrebbe essere dannoso.