	Politecnico di Milano, Facoltà di Ingegneria Industriale, Prof. Boracchi **INFORMATICA B** Prova in itinere del 6 Febbraio 2014		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA
			Spazio riservato ai docenti

- Il presente plico contiene 3 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 1 ora e 15 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!) **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.
- **Non è permesso consultare temi d'esame degli anni precedenti**
- 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)

1) Si implementi in MATLAB una funzione **combinaMatrici** che svolga le seguenti operazioni:

- Riceve in ingresso due matrici **A** e **B** aventi le stesse dimensioni.
- Produce una terza matrice C ottenuta da **A** e **B** secondo la seguente regola
    - Nelle posizioni (r, c) in cui **B**(r, c) è minore del minimo di **A**, **C** assume il valore di **A**(r, c), altrimenti assume il valore di **B**(r, c)

Per minimo di **A** si intende il più piccolo di tutta la matrice **A.**

Si scriva la funzione **combinaMatrici** sfruttando le caratteristiche di sintesi del linguaggio Matlab (possibilmente evitando l'utilizzo di cicli).

Esempio input:
A = [2 2; 4 3]
B = [3, 5; 5,-1]

Rispettivo output:
C = [3 5; 5, 3]

1) Si scriva una funzione **combinaMatriciSup** simile alla precedente, ma con le seguenti differenze. La funzione **combinaMatriciSup** riceve in ingresso le due matrici **A** e **B**, e un function handle **f**. La funzione restituisce una matrice **C** così definita
    - Nelle posizioni (r,c) in cui **B**(r,c) è minore di **f** applicato al minimo di **A**, **C** assume il valore **A**(r,c), altrimenti assume il valore di **B**(r,c)

Esempio input:
A = [2 2; 4 3];
B = [3, 5; 5,-1];
f = @(x)(x.^2);

Rispettivo output:
C = [2 5; 5, 3]

Si scriva una chiamata a **combinaMatriciSup** passando come argomenti:
A: la matrice identità 3x3;
B: una matrice di diemensione 3x3 contentente numeri interi casuali tra -10 e 10;
f: la funzione che restituisce la differenza tra seno e coseno di un numero.

Soluzione

```
function C = combinaMatrici(A, B)
        C = B;
        C(min(A(:)) > B)=A(min(A(:)) > B)
end


function C = combinaMatriciSup(A, B, f)
        C = B;
        minA = f(min(A(:)));
        C(minA > B) = A(minA > B)
end


A = eye(3);
B = round(rand(3)*20-10);
f = @(x)(sin(x)-cos(x));

combinaMatriciSup(A, B, f)
```

Esercizio 2 (6 punti)

Siano date le seguenti funzioni:

```
function [qua] = paperone(a, b)
if b==1 || b==0
   qua = false;
else
   if (mod(a, b) == 0) && a~=b
      qua = true;
   else
      qua = paperone(a, b-1);
   end
end
end

function [x] = paperino(aaargh)
x = true;
if paperone(aaargh, aaargh) ~= 1;
   x=false;
   return
end
end
```

1. Si dica che proprietà devono soddisfare **a** e **b** perché paperone(a,b) sia true.

2. Si dica che proprietà deve soddisfare **k** (numero intero) perché paperino(**k**) sia true.

3. La funzione **paperone()** dà luogo ad un numero finito di chiamate ricorsive qualunque siano i numeri passati come parametri attuali **a**, **b**? Si motivi la risposta e, se necessario, si modifichi opportunamente la funzione per evitare chiamate infinite.

Soluzione

1. **paperone(a, b)** controlla se esiste un divisore di **a** che sia minore di b e che sia diverso da **a** e **1**

2. Indica se il numero **k** è un numero non primo.

3. La funzione **paperino()** da luogo ad una sequenza di chiamate infinite quando **b** è negativo o quando almeno uno degli argomenti è non intero. Per evitare questa situazione basta modificare la condizione **b==1 || b==0** con **b<=1**

Si noti che la chiamata con argomenti interi tali per cui **b > a** non da luogo ad una sequenza di chiamate infinite perché si raggiunge il caso base **b == 1**.

Esercizio 3 (4 punti)

Un sistema basato su microprocessore, senza nessuna cache, ha un tempo di accesso alla memoria di 100ns.

1. Quale delle seguenti configurazioni *con memoria cache* può migliorare le performance del sistema? Si motivi adeguatamente la risposta

Configurazione	Hit time	Hit rate	Miss-penality
1	10ns	0.2	150ns
2	3ns	0.2	140ns
3	12ns	0.4	150ns

2. Si scriva una funzione MATLAB **selezionaCache** che prende in ingresso una matrice avente 3 colonne e contenente, in ogni riga, le specifiche di una tipologia di memoria cache: hit time nella prima colonna, hit rate nella seconda e miss-penality nella terza. La funzione **selezionaCache** deve restituire l'indice della riga corrispondente alla configurazione con migliori performance.

Soluzione

1. tempi di accesso medio delle tre soluzioni
1) 10ns * 0.2 + 150ns *0.8 = 122ns
2) 3ns * 0.2 + 140ns * 0.8 = 112.6ns
3) 12ns* 0.4 + 150ns * 0.6 =94.8ns (migliore)

2.

```
function idx = selezionaCache(A)
   times = A(:,1) .* A(:,2) + A(:,3) .* (1-A(:,2));
   [~, idx] = min(times);
end
```