	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione INFORMATICA B Appello 15 Settembre 2017		COGNOME E NOME										
			MATRICOLA										
Spazio riservato ai docenti													
<table border="1" style="margin-left: auto;"> <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> <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.**
- Il tempo a disposizione è di 1 ore e 30 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 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 ricaricare al momento della consegna!).
- **È vietato utilizzare telefoni o pc.** Chi tenti di farlo vedrà annullata la sua prova.
- È ammessa la consultazione di libri e appunti.
- **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 (10 punti)

- A. Scrivere una funzione MATLAB $[x, y] = \text{somme}(n)$, che riceve in ingresso un intero positivo n e restituisce due vettori x e y , contenenti tutte le coppie di interi (x, y) la cui somma è pari ad n .

Note: L'ordine degli interi restituiti in ciascuna coppia è irrilevante: se viene restituita la coppia (x, y) non occorre restituire anche la coppia (y, x) .

Esempi

somme(8) restituisce $x = [1, 2, 3, 4]$ e $y = [7, 6, 5, 4]$

somme(11) restituisce $x = [1, 2, 3, 4, 5]$ e $y = [10, 9, 8, 7, 6]$

- B. Scrivere inoltre uno script che, dopo aver chiesto all'utente di fornire un valore val maggiore di 3, richiama la funzione `somme` un numero di volte compreso tra val e 3 e genera un grafico che mostra i risultati ottenuti dalle chiamate.

Soluzione

```
function [x,y] = somme(n)
x = [1:1:n/2];
y = n - x;
```

```
%script
val = input('inserisci un valore positivo maggiore di tre: ');
hold on;
while val > 3
    [x, y] = somme(val);
    plot(x, y);
    val = val-1;
end
hold off
```

Esercizio 2 (10 punti)

La compagnia aerea SuperCheap verifica ogni mese che il tempo di volo dei suoi piloti non sia superiore a 3000 minuti. A questo scopo utilizza le seguenti strutture dati:

```
typedef struct
{
    // DATI GIA' INSERITI
    int id; //id numerico univoco
    char nome[100]; //nome e cognome
    int nvoli; //numero voli nel mese
    int voli[100]; //id dei voli effettuati dal pilota nel mese

    //DA CALCOLARE
    int tempo_volo; //tempo di volo in minuti nel mese
} dati_pilota;

typedef struct
{
    int id; //id numerico univoco
    char from[4]; // partenza
    char dest[4]; // destinazione
    int durata; //durata del volo in minuti
} dati_volo;

dati_pilota p[50]; // dati dei piloti per Ottobre 2017
dati_volo v[215]; // dati dei voli per Ottobre 2017
```

Dove le variabili `p` e `v` contengono rispettivamente i dati dei 50 piloti della compagnia e dei 215 voli previsti per il mese di Ottobre 2017. **Entrambe le variabili contengono già tutti i dati, ad eccezione del tempo di volo nel mese di ciascun pilota.**

- Si scriva un frammento di codice in C per calcolare il tempo di volo nel mese di ciascun pilota ed aggiornare di conseguenza il valore del campo `tempo_volo` di ciascuno di essi.
- Attraverso un frammento di codice C, si calcoli il numero `n` di piloti che nel mese voleranno più di 3000 minuti e si memorizzino gli `n` identificativi (`id`) in una opportuna variabile/struttura dati.

Nota. Non è necessario riportare il codice per acquisire i dati delle variabili `p` e `v`; è sufficiente riportare i frammenti di codice C richiesti ai punti A e B (insieme ad eventuali dichiarazioni di variabili utilizzate).

Soluzione

```
A.
int i, j, k;
for (i=0; i<50; i++)
{
    p[i].tempo_volo = 0;
    for (j=0; j<p[i].nvoli; j++)
        for (k=0; k<215; k++)
            if (p[i].voli[j] == v[k].id)
                p[i].tempo_volo += v[k].durata;
}
```

B.

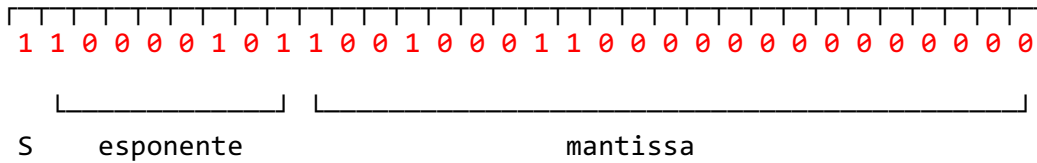
```
int n=0; //numero dei piloti che volano più di 3000 minuti.
int sel[50]; //array per contenere gli id di chi vola più di 3000 minuti.
for (i=0; i<50; i++)
    if (p[i].tempo_volo > 3000)
    {
        sel[n] = p[i].id;
        n++;
    }
}
```

Esercizio 3 (3 punti)

Si consideri il numero negativo e razionale -100.375

Riportare nello spazio seguente la sua codifica secondo lo standard IEEE a precisione singola (riportare di seguito anche i calcoli effettuati).

Soluzione



S: 1

$$100 = 4+32+64 \rightarrow 1100100$$

$$0.375 * 2 = 0.75$$

$$0.75 * 2 = 1.5$$

$$0.5 * 2 = 1.0$$

$$0.375 \rightarrow 0.011$$

$$1100100.011 = 1.100100011 * 2^6$$

$$E = 6+127 = 133 = 128+4+1 = 10000101$$

S: 1

M: 1001000110 0000000000 000

E: 10000101

Esercizio 4 (5 punti)

Un sistema ha una memoria con le seguenti caratteristiche: memoria fisica e virtuale indirizzabile entrambe di 32 Mbyte e pagine di memoria virtuale di dimensione 16Kbyte.

Rispondere alle seguenti domande giustificando le risposte:

- A. Quale è la struttura dell'indirizzo virtuale e di quello fisico, e la lunghezza dei campi che li costituiscono?
- B. Un consulente afferma che essendo la memoria fisica e la memoria virtuale della medesima dimensione, non potranno mai esserci pagine **non residenti** in memoria e quindi non è necessario prevedere uno spazio sul disco fisso per ospitare le pagine non residenti. Siete d'accordo? Argomentare in maniera adeguata la propria risposta.

Soluzione

- A. 32 Mbyte di memoria indirizzabile \rightarrow 25 bit di indirizzo
Pagina di 16Kbyte = 2^{14} byte \rightarrow 14 bit offset

Quindi si ha,

Indirizzo virtuale: NPV: 11 bit, offset: 14 bit

Indirizzo fisico: NPF: 11 bit, offset: 14 bit

- B. L'affermazione è errata, dal momento che la memoria fisica è condivisa da più processi, quindi anche se la memoria virtuale del **singolo** processo non può eccedere la dimensione della memoria fisica, l'insieme dei processi caricati in un dato momento potrebbero avere delle richieste di memoria che superano la dimensione della memoria fisica.