

	Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione FONDAMENTI DI INFORMATICA Appello 26 Gennaio 2018		COGNOME E NOME					
	RIGA	COLONNA	CODICE PERSONA					
			Spazio riservato ai docenti <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; height: 20px;"></td> </tr> </table>					

- Il presente plico contiene 5 esercizi e **deve essere debitamente compilato con cognome e nome, codice persona.**
- 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 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 telefoni o pc.** Chi tenti di farlo vedrà annullata la sua prova.
- **Non 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 (8 punti)

- A. Definire il tipo di dato `rotta` che consente di rappresentare una rotta aerea, specificando: codice aeroporti di partenza e di destinazione (i codici degli aeroporti sono 3 lettere) e distanza in miglia della rotta (numero intero).
- B. Scrivere un programma C che dichiari un array di tipo `rotta` di dimensione 100 e ne acquisisca da tastiera il contenuto; quindi, chieda di inserire da tastiera il codice di un aeroporto di partenza e stampi a video il codice dell'aeroporto di destinazione più vicino in miglia.
Nel caso ci fossero più aeroporti di destinazioni alla stessa distanza minima, è sufficiente stamparne uno. Se invece non esistesse alcun elemento dell'array di tipo `rotta` con l'aeroporto di partenza richiesto, si stampi un messaggio di errore.

Soluzione

```
#include <stdio.h>
#include <string.h>

#define N 100

typedef struct
{
    char part[4];
    char dest[4];
    int dist;
} rotta;

int main()
{
    rotta dati[N];
    char part[4];
    int i,imin;
    for (i=0; i<N; i++)
    {
        scanf("%s%c",dati[i].part);
        scanf("%s%c",dati[i].dest);
        scanf("%d%c",&dati[i].dist);
    }

    scanf("%s",part);

    imin = -1;
    for (i=0; i<N; i++)
    {
        if (strcmp(dati[i].part,part)==0 && (imin===-1 || dati[i].dist < dati[imin].dist))
            imin = i;
    }

    if (imin===-1)
        printf("Aeroporto di partenza (%s) non trovato\n",part);
    else
        printf("Aeroporto più vicino: %s\n",dati[imin].dest);

    return 0;
}
```

Esercizio 2 (9 punti)

Scrivere una funzione C che riceve in ingresso due teste di liste contenenti numeri interi (scrivere anche le eventuali dichiarazioni di tipo) e restituisce 1 se la prima lista è strettamente contenuta nell'altra (cioè tutti gli elementi della prima lista appartengono anche alla seconda) o 0 altrimenti. **Ciascuna lista non contiene elementi ripetuti e i suoi elementi non sono ordinati.**

Soluzione

```
struct nodo
{
    int el;
    struct nodo *next;
};

typedef struct nodo *lista;

int contiene(lista l1, lista l2)
{
    lista cur;
    while (l1!=NULL)
    {
        cur = l2;
        while(cur!=NULL && cur->el != l1->el)
            cur = cur->next;

        if (cur==NULL)
            return 0;

        l1 = l1->next;
    }
    return 1;
}
```


Esercizio 4 (8 punti)

Implementare in C la seguente funzione ricorsiva:

```
int resto(int x, int y)
```

La funzione riceve in ingresso due numeri interi x e y (con $x \geq 0$ e $y > 0$), quindi calcola e restituisce il resto della divisione intera x/y . **Nell'implementazione è possibile utilizzare solo gli operatori relazionali di confronto ($>$, $<$, \geq , \leq , $==$ e $!=$), somma (+) e sottrazione (-). Non è possibile utilizzare altri operatori ($/$, $*$, $\%$) o altre funzioni di libreria.**

Soluzione

```
int resto(int x, int y)
{
    if (x < y)
        return x;
    else
        return resto(x-y, y);
}
```

Esercizio 5 (4 punti)

Utilizzando il linguaggio macchina introdotto a lezione, scrivere un programma che legga dal nastro di ingresso un numero intero x e una sequenza di valori interi terminata da uno 0. Il programma dovrà quindi stampare sul nastro di uscita tutti i valori della sequenza letta in ingresso che sono, in **valore assoluto**, maggiori di x .

Suggerimento: per calcolare il valore assoluto di un numero è sufficiente moltiplicarlo per -1 se negativo.

Soluzione

```
1.      READ
2.      STORE 101
3.      READ
4.      STORE 102
5.      BEQ 13
6.      BG 8
7.      MULT= -1
8.      SUB 101
9.      BLE 3
10.     LOAD 102
11.     WRITE
12.     BR 3
13.     END
```