# Laboratorio Informatica A

## Problema 1

Scrivere un programma che chiede all'utente di inserire un numero positivo (nel caso in cui il numero non è positivo ripetere inserimento) e verifica se è perfetto, stampando un messaggio all’utente.

Se non è perfetto stampa un messaggio per dire se è abbondante o difettivo e richiede un secondo numero intero positivo e controlla se ed sono amici. Si stampa a schermo il risultato di questo controllo.

**Definizioni:**

Un numero è perfetto se corrisponde alla somma dei suoi divisori, escluso sè stesso (es. 6 è perfetto 1 + 2 +3 = 6)

Un numero è abbondante se è > della somma dei suoi divisori (es 15 è abbondante 1 + 3 +5 < 15), altrimenti difettivo (es 12 è difettivo, 1+2+3+4+6 > 12)

Due numeri a, b sono amici (o amicabili) se la somma dei divisori di a è uguale a b e viceversa (es 220 e 284)

Consiglio per la soluzione, si sviluppino le seguenti funzioni:

**int sommaDivisori(int);  
int controllaSePerfetto(int);  
int controllaSePerfettoRef(int, int\*);** // da fare con il passaggio per riferimento, se speigato a lezione

**int controllaSeAmici(int, int);  
int leggiInteroPositivo();**

## Problema 2

Si scriva un frammento di codice legga da input le coordinate x e y di una struct “punto” cosi definita:

typedef struct {

float x;

float y; } Punto;

si sviluppino ed invochino in un main le seguenti funzioni:

* **void stampaPunto(Punto p)** che effettui la stampa di un punto passato come parametro
* **Punto acquisisciPunto()** che crea un punto dopo aver chiesto le coordinate all’utente.   
  Si modifichi (dopo lezione 27/10) in **void acquisisciPunto(Punto \* p)**
* **float calcolaNorma(Punto p)** che restituisce la distanza del punto dall’origine
* **float calcolaDistanza(Punto p1, Punto p2)** che calcola la distanza tra il punto p1 e p2

Si scriva infine una funzione “**stampaLunghezza**” che riceve un vettore di punti e restituisce la lunghezza della linea spezzata composta dai punti nel vettore (nell’ordine in cui sono nel vettore). La funzione avrà prototipo:

**float lunghezza(Punto linea[])**

## Problema 3

Si consideri questa struttura che può contenere i dati di una semplice rubrica telefonica

**#define MAXSTRLEN 101**

**typedef struct {**

**char cognome[MAXSTRLEN];**

**char nome[MAXSTRLEN];**

**char cellulare[MAXSTRLEN];**

**} RECAPITO;**

Si limiti a un massimo di 10 il numero di recapiti memorizzabili.

Si scriva quindi un programma che, tramite un apposito menù, consenta l'inserimento di un recapito telefonico, la visualizzazione dei recapiti inseriti e l'eliminazione (con compattamento) di uno specifico recapito.