AA 2021-22

FONDAMENTI DI INFORMATICA

L A B O R A T O R I O

[prof.ssa RAFFAELA MIRANDOLA]

POLITECNICO
MILANO 1863

Informazioni Logistiche

LABS:

Online Giovedi 9:15-12:15

GIO 14/10/21	Lab1	9:15-12:15	AULA VIRTUALE
GIO 28/10/21	Lab2	" "	" "
GIO 18/11/21	Lab3	" "	" "
GIO 02/12/21	Lab4	" "	" "
GIO 16/12/21	Lab5	" "	" "

Tutors:

sofia.gautieri@mail.polimi.it
agnese.taluzzi@mail.polimi.it
pietro1.lodi@mail.polimi.it

AULA VIRTUALE:
https://politecnicomilano.webex.com/meet/gianenrico.conti

contatti | mail | link

Responsabili di Laboratorio

Ing. *Gian Enrico* Conti gianenrico.conti@polimi.it

Sito web del corso

https://webeep.polimi.it/course/view.php?id=1128

Cicli, array, stringhe, struct

Iniziamo con 4 esercizi di “riscaldamento”, da eseguire in pochi minuti.

Seguiranno altri esercizi più corposi, quindi tutto va pensato e realizzato rispettando i tempi.

Esercizio 3.1 “warm-up”

Si scriva un programma che legga da input di 10 numeri interi ed un intero X compreso tra 0 e 9.

Il programma deve stampare la somma dei numeri nelle posizioni minori di X, e il prodotto dei numeri in posizioni
successive a X.

Esercizio 3.2 “warm-up”

Si scriva in programma che legga una stringa “C” e stampi in output il numero di vocali presenti, sia maiuscole che minuscole.

Suggerimenti

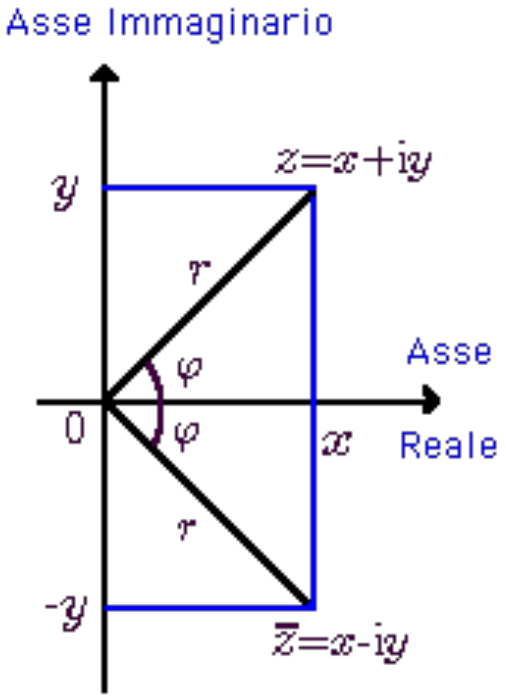
- **Chi vuole può usare il costrutto di selezione multipla `switch`.**

Esercizio 3.3 "warm up"

Si definisca uno *struct* opportuno a rappresentare i numeri complessi. (*Un numero complesso è definito da una parte reale* (Re) *e una immaginaria* (Im).)

Si scriva un programma che:

- Legga due numeri complessi `n1` e `n2` e ne calcoli la somma (che è a sua volta un numero complesso)
- Verifichi e dica se `n1` e `n2` se sono uguali in fase.



Esercizio 3.4

Usando *struct* ed *array*, si scriva un programma C che legga un array (chiamato ***valori***) di MAX numeri complessi e stampi il più **grande** in modulo ed il **minore** in fase

Non serve utilizzare sqrt!

Hint: simile ad un problema di ricerca del massimo/min.

Hint 2: x calcolare arcotangente: esiste atn(x)

(Vedi: nella shell digitare:

```
man atan
```

X help

Hint3: gcc -lm ……..s (per le lib matematiche..)

Esercizio 3.5

Si definisca uno *struct* Exam che comprenda:

* nome dell' esame
* voto [1;30]

Si definisca uno *struct* Student che comprenda:
* codice_persona
* array degli esami

Si scriva un programma che:
1) legga un array di 10 (MAX) studenti
2) stampi tutti i voti (e relativo nome dell'esame) di uno studente dato un codice_persona in input
3) *advanced:* stampare lo studente con la media più alta

Nota: è ammesso scrivere il programma sia utilizzando sottoprogrammi che con il solo main()

Esercizio 3.5 casi di test.

Usate questi casi di test:

1' studente:

```
            "244433", // matr.
             {   // esami
                 { "Analisi I", 30},
                 { "Fisica I", 23},
                 { "Informatica B", 18},
             }
        }
```

2' studente:

```
        {
            "45665445", // matr.
            {   // esami
                { "Analisi II", 30},
                { "Fisica II", 23},
                { "Elettronica", 24},
            }
        }
```

3' studente:

```
        {
            "884489",// matr.
            {   // esami
                { "Analisi III", 22},
                { "Fisica III", 19},
                { "Elettrotecnica", 27},
            }
        }
    }
```

Esercizio 3.6

Si scriva un sottoprogramma "asciiToInt" che converta una stringa nel corrispondente valore intero.

Se la stringa non contiene la rappresentazione di un numero,
Il valore restituito dal sottoprogramma e' -1.

Si scriva anche il main() di test che legga da input la stringa e stampi il valore restituito dal sottoprogramma.

Esempio: s = “1234”
output: 1234

Nota:
(Nelle lib. "C" ha nome: ***"atoi", che NON va usata...ma riscritta!***)

Esercizio 3.7 ("check_array")

Scrivere un **sottoprogramma** che riceve come parametri due array di interi di egual dimensione, e la loro dimensione.

Il sottoprogramma verifica se, pur di fare uno scorrimento circolare del secondo array, i due array siano identici.

Nello scorrimento circolare, la cifra che si trova in ultima posizione diventa quella in prima posizione dopo lo scorrimento.

In caso affermativo il sottoprogramma restituisce quante posizioni deve essere fatto scorrere il secondo array verso destra.

In caso negativo, restituisce -1.

Scrivere un **programma** che acquisisce i dati per popolare due array monodimensionali di 10 interi ciascuno.

Il programma invoca il **sottoprogramma** sopra definito e visualizza il risultato.

Esercizio 3.7 esempio

Esempio:

se il sottoprogramma riceve i seguenti due array:
a = [0 2 4 8 2 1 7 8 2 9]
b = [1 7 4 8 2 7 2 6 1 6]

Il sottoprogramma restituisce -1

se il sottoprogramma riceve i seguenti due array:
a = [5 4 9 8 3 0 1 9 3 4]
b = [9 3 4 5 4 9 8 3 0 1]

Il sottoprogramma restituisce 7

Esercizio 3.8 ("spezzata")

Un punto nel piano cartesiano è definito dalla coppia di coordinate intere x e y: si definisca l'opportuno tipo di dato **point_t**.

Data una matrice NxN di punti, i punti della diagonale, quelli di ogni riga e quelli di ogni colonna definiscono linee spezzate (ciascuna di N-1 lati).

Si definisce *regolare* una matrice dove la lunghezza della spezzata definita dalla diagonale ha lunghezza maggiore della lunghezza di tutte le spezzate definite dalle righe e dalle colonne della matrice.

Si definisca un sottoprogramma ***regolare*** che ricevuta in ingresso una matrice di punti e qualsiasi parametro ritenuto strettamente necessario,
restituisce 1 se la matrice è regolare,
0 altrimenti.

Esercizio 3.8

Ipotesi di lavoro:

Si supponga che esista una direttiva
define N ...

che indica la dimensione utilizzata nella dichiarazione della matrice quadrata passata al sottoprogramma.

Infine si suggerisce di realizzare un apposito sottoprogramma di ausilio

float dist(point_t p1, point_t p2)

che calcola e restituisce la distanza euclidea tra due punti.