



POLITECNICO
MILANO 1863

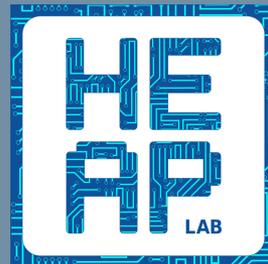
Informatica ed Elementi di Informatica Medica

A.A. 2017-18

Laboratorio n°2

Dott. Michele Zanella

Ing. Gian Enrico Conti



- Contatti:
 - gianenrico.conti@mail.polimi.it
 - michele.zanella@polimi.it
 - [HEAP Lab](#) – Campus Leonardo, via Golgi 39, Edificio 21, Piano 1, Ufficio 4, +39 02 2399 **9613**
(mandatemi una mail per accordarci su giorno e ora)
- Sito web del laboratorio:
 - <http://zanella.faculty.polimi.it/teaching/informatica-ed-elementi-di-informatica-medica/>
- **Nota per le mail:**
Oggetto: *[IEIM] il vostro oggetto*

- Calendario laboratori

Data	Orario	Squadra	Aula	Resp.	Programma
16/03/18	14:15-17:15	A	L26.14	Conti	Lab 1
19/03/18	15:15-18:15	B	L26.14	Zanella	
23/03/18	14:15-17:15	A	L26.14	Conti	Lab 2
26/03/18	15:15-18:15	B	L26.14	Zanella	
06/04/18	14:15-17:15	A	L26.14	Conti	Lab 3
09/04/18	15:15-18:15	B	L26.14	Zanella	
13/04/18	14:15-17:15	A	L26.14	Conti	Lab 4
16/04/18	15:15-18:15	B	L26.14	Zanella	
20/04/18	14:15-17:15	A	L26.14	Conti	Lab 5
23/04/18	15:15-18:15	B	L26.14	Zanella	

Esercizio 2.1: Ricerca sequenziale

Scrivere un programma che, dato un vettore di n numeri, implementi una funzione (dichiarata come segue) per ricercare un valore al suo interno.

```
int ricerca_seq(int insieme[], int n, int x);
```

La funzione può ritornare il numero di ricorrenze del carattere oppure l'indice del vettore che contiene il numero da ricercare

Esercizio 2.2: Crivello di Eratostene



Scrivere un programma che implementi il [crivello di Eratostene](#), antico algoritmo per il calcolo dei primi n numeri primi.

Esercizio 2.3: Ordinamento – Bubble sort

Scrivere un programma che, dato un vettore non ordinato di n numeri, implementi l'algoritmo Bubble sort per ordinare i valori in ordine crescente seguendo.

Bubble sort: Ogni coppia di elementi adiacenti viene comparata e invertita di posizione se sono nell'ordine sbagliato. L'algoritmo continua nuovamente a ri-eseguire questi passaggi per tutta la lista finché non vengono più eseguiti scambi, situazione che indica che la lista è ordinata.

Esercizio 2.4: Riflessione stringa

Scrivere un programma che, data una stringa, calcoli la sua riflessa.

Esempio:

"roma" -> *"amor"*

Esercizio 2.5: Stringa palindroma

Scrivere un programma che, data una stringa (senza spazi), verifica se è palindroma.

Esempio:

"anna" -> è palindroma

"pippo" -> non è palindroma

Esercizio 2.6: Confronto tra stringhe

Scrivere un programma che, date due stringhe, implementi una funzione che verifichi se sono uguali o in caso negativo quale viene prima in ordine alfabetico.

```
int string_cmp(char stringa1[], char stringa2[]);
```

La funzione deve ritornare:

- 0: le stringhe sono uguali
- -1: la `stringa1` è minore della `stringa2`
- 1: la `stringa2` è maggiore della `stringa1`

Esercizio 2.7: `getint()/putint()`

Scrivere un programma che implementi le funzioni per leggere da input e scrivere in output degli interi.

```
int getint();
```

```
int putint(int n);
```

Hint:

Sfruttare le funzioni di libreria `getchar()/putchar()` viste a lezione.

Ulteriori esercizi

- Re-implementare gli esercizi 2.1, 2.2, 2.3, 2.4, 2.5 permettendo all'utente di inserire i valori tramite le funzioni di I/O getchar/getint.
- Re-implementare gli esercizi 2.1, 2.2, 2.3 passando i numeri richiesti come argomenti del main.