

*Università degli Studi di Trieste*

**Corsi di ingegneria industriale e navale**

**Esercitazioni di Fondamenti di Informatica**

**Giacomo Strangolino**

mailto: [giacomo.strangolino@elettra.trieste.it](mailto:giacomo.strangolino@elettra.trieste.it)

**Tel: 328 3237959**

**<http://www.giacomos.it>**

## Lezione 2

- Questione ricevimento;
- contatto online su gmail/gtalk:  
[delleceste@gmail.com](mailto:delleceste@gmail.com);
- *Homework*
- *Kernighan, Ritchie: ANSI C Programming Language*

## Lezione 2

Scrivere alcuni semplici programmi atti a:

1. fare l'“eco” dello *standard input* nello *standard output*;
2. immesso un vettore da tastiera, calcolare il minimo valore, il massimo e la media;
3. immesse due stringhe da tastiera, dire se sono uguali oppure qual è la più breve e di quanti caratteri;
4. immessa una stringa, contare le occorrenze di ogni cifra, degli spazi e di tutti gli altri caratteri e produrre un istogramma verticale;

## Lezione 2

# Altri esercizi (II)

Scrivere alcuni semplici programmi atti a:

1. ordinare un vettore di interi (algoritmo *bubble sort*);
2. data una matrice quadrata di dimensione  $N$ , estrarre la diagonale in un vettore e la stampi. La matrice può essere passata in qualche modo da tastiera oppure inizializzata all'interno del programma.

## Lezione 1

# Homework

1. Si modifichi il programma della diagonale della matrice in modo che gli elementi vengano immessi dall'utente;
2. Ottimizzare il bubble sort osservando che (1) quando in una data iterazione non avviene più uno scambio, l'algoritmo può terminare e (2) se una data iterazione non sposta nessun elemento di posizione maggiore di un dato valore  $i$ , allora si può facilmente dimostrare che nessuna iterazione successiva eseguirà alcuno scambio in posizioni successive a tale valore  $i$ .
3. Si modifichi il programma dell'istogramma verticale in modo che l'istogramma prodotto sia orizzontale.
4. Si scriva un programma che, data una stringa in ingresso, la riproponga sullo *standard output* mandando a capo ogni  $n$  parole, dove  $n$  è scelto dall'utente.

## Lezione 2

# Homework (II)

1. Esercizi delle provette degli anni precedenti!
2. Si scriva un programma che, data una stringa in ingresso, dica se questa è palindroma. Ad esempio “O MORDO TUA NUORA O ARO UN AUTODROMO” sarà palindroma (senza spazi).
3. Scrivere un programma che concateni due stringhe in ingresso.
4. Scrivere un programma che elimini tutte le maiuscole da una stringa letta da tastiera.
5. Scrivere un programma “mytolower” che trasformi in minuscole le eventuali lettere maiuscole in una stringa letta da tastiera.
6. Scrivere un programma che, letta una stringa  $s$  e un intero  $n$ , stampi  $s$  mandando a capo ogni  $n$  parole. I confini delle parole siano determinati da uno spazio o da un carattere di tabulazione ('\t'). Per semplicità, si supponga che l'utente immetta un solo spazio o *tab* alla volta per separare le parole.

# Lezione 2

## Standard ASCII table

ASCII Decimal Equivalent Character Chart								
ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☹	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	\$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	■	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♪	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	▲	DLE	048	0	080	P	112	p
017	▼	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	☐	SYN	054	6	086	V	118	v
023	↕	ETB	055	7	087	W	119	w
024	↕	CAN	056	8	088	X	120	x
025	↕	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	☐

## Lezione 2

# Algoritmo Bubble Sort

ogni coppia di elementi adiacenti della lista viene comparata e se essi sono nell'ordine sbagliato vengono invertiti. L'algoritmo scorre poi tutta la lista finché non vengono più eseguiti scambi, situazione che indica che la lista è ordinata.