

 POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Prima esercitazione: algoritmi e tipi di dato

Matteo Ferroni
matteo.ferroni@polimi.it

29/09/2017



POLITECNICO
DI MILANO



Chi sono



Matteo Ferroni

- Ph.D. @ NECSTLab w/ Santa
- Ho lavorato su modelli di potenza per sistemi energy-aware
- Esercitatore di questo corso
- Web site: **matteo.ferroni.me**
- Email: **matteo.ferroni@polimi.it**

b o t
t e g
a 5 2.it

Agenda

- Un programma in C
- Algoritmi: esempio del DVD
- I tipi di dato
- Caratteri e codifica ASCII

Un programma in C

parte dichiarativa **globale**

inclusione librerie / per poter invocare funzioni utili (i/o, ...) /
dichiarazione di **variabili globali e funzioni**

```
int main ( ) {
```

parte dichiarativa **locale**

dichiarazione di **variabili locali**

```
istruzione 1; / tutti i tipi di operazioni, e cioè: /  
istruzione 2; / istr. di assegnamento /  
istruzione 3; / istr. di input / output /  
istruzione 4; / istr. di controllo (condizionali, cicli) /
```

```
...
```

```
istruzione N;
```

parte **esecutiva**

```
return 0;
```

```
}
```

Un programma in C

Ma prima di pensare al codice...

**RAGIONATE
SULL'ALGORITMO!**



Dove lo cerco?
In quale *settore*?

Novita'

Azione

...

Indicatori di settore



Trovato il settore, come
trovo il mio DVD?

Ordine alfabetico



1. **Acquisisci** il nome del settore (s)
del DVD (d)
 2. Vai al settore (s)
 3. **Cerca** (t)
 4. Prendi il DVD (d)
- **Acquisisci** e **Cerca** sono anche loro *algoritmi*



- Contesto
 - Sei nel settore (s) corretto
 - Nel settore sono presenti diversi DVD
 - Il settore puo' essere visto come un insieme di DVD: $s = \{d_0, d_1, \dots, d_n\}$
- Problema
 - Devo trovare il DVD (d_t) tra tutti i DVD presenti nel settore (s)
 - **Aiuto:** il settore e' ordinato
 - I DVD sono ordinati alfabeticamente



- Noti
 - $s = \{d_0, d_1, \dots, d_n\}$, ord. alfabeticamente
 - DVD cercato = d_t

1. Ci sono DVD?

- Allora
 1. Leggo il titolo del primo DVD in s
 2. Se il titolo e' il titolo cercato
Allora concludo la ricerca con successo
Altrimenti passo al DVD successivo
- Altrimenti concludo la ricerca con esito negativo

- Cosa succede se il settore e' vuoto?
- Come *funziona* la mia ricerca?
 - Se in s vi sono 10000 DVD e io cerco Zorro?
 - Scenario ancora peggiore
 - Voglio cercare il numero del mio professore di IEIM (Santambrogio) nella rubrica telefonica di Milano..
- Esistono diversi modi per risolvere un problema



- Noti
 - $s = \{d_0, d_1, \dots, d_n\}$, ord. alfabeticamente
 - DVD cercato = d_t

1. Ci sono DVD?

- Allora
 1. **Leggo** il titolo del DVD (d_x) nel *mezzo* di s
 2. **Se** il titolo di d_x e' il titolo cercato
 - **Allora** concludo la ricerca con successo
 - **Altrimenti se** $d_x < d_t$
 - » **allora** ricomincio da 1 considerando la meta' superiore di s
 - » **Altrimenti** ricomincio da 1 considerando la meta' inferiore di s
- **Altrimenti** concludo la ricerca con esito negativo



Graficamente: Noti



d_t

$$s = \{d_0, d_1, \dots, d_{11}\}$$

d_0

d_1

d_2

d_3

d_4

d_5

d_6

d_7

d_8

d_9

d_{10}

d_{11}





Leggo d_x e lo confronto con d_t



$$d_x = d_6$$



=



?



NO



<





Ricomincio *sulla meta' superiore*



d_6

d_7

d_8

d_9

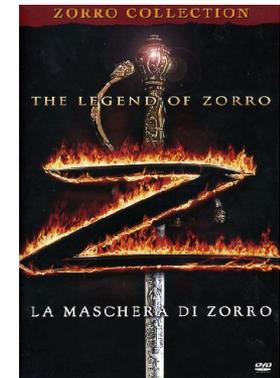
d_{10}

d_{11}

d_t



$$d_x = d_9$$





Leggo d_x e lo confronto con d_t



$$d_x = d_9$$

$$d_9 = d_t ?$$



SI

Termino con successo la mia ricerca comprando d_9



- Problema
 - Richiedi all'utente la sua altezza in centimetri e mostrala a video in metri
- Pseudocodice
 1. Scrivi "quanto sei alto?"
 2. Leggi altezzacm
 3. $\text{Altezzam} = \text{alteccacm} / 100$
 4. Scrivi "sei alto: altezzam"



- Pseudocodice

1. Scrivi “quanto sei alto?”
2. Leggi altezzacm
3. Altezzam = alteccacm/100
4. Scrivi “sei alto: altezzam”

```
#include <stdio.h>

int main () {
    int altezzacm;
    int altezzam;

    printf("Quanto sei alto? |");
    scanf("%d",&altezzacm);
    altezzam = altezzacm/100;
    printf("Sei alto %d m\n",altezzam);

    return 0;
}
```



Perché %d!



```
#include <stdio.h>

int main () {
    int altezzacm;
    int altezzam;

    printf("Quanto sei alto? |");
    scanf("%d",&altezzacm);
    altezzam = altezzacm/100;
    printf("Sei alto %d m\n",altezzam);

    return 0;
}
```

Tra “” ci si aspetta una sequenza di caratteri

Provate ad inserire
-1347

Si sono premuti 6 caratteri:

-, 1, 3, 4, 7, enter

Abbiamo quindi inserito una sequenza di caratteri!

%d dice al calcolatore di interpretare questa sequenza come un intero

- Pseudocodice

1. Scrivi “quanto sei alto?”
2. Leggi altezzacm
3. $\text{Altezzam} = \text{alteccacm}/100$
4. Scrivi “sei alto: altezzam”

```
#include <stdio.h>

int main () {
    int altezzacm;
    int altezzam;

    printf("Quanto sei alto? |");
    scanf("%d",&altezzacm);
    altezzam = altezzacm/100;
    printf("Sei alto %d m\n",altezzam);

    return 0;
}
```

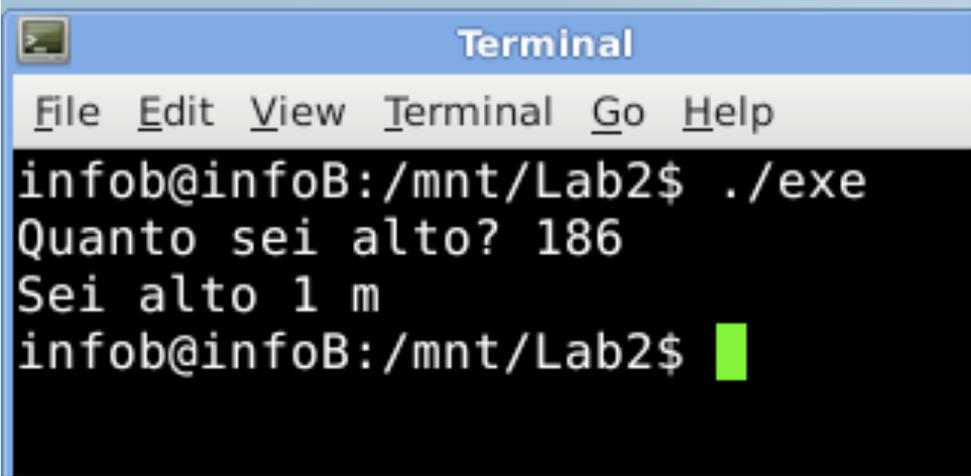
Un primo errore

```
#include <stdio.h>

int main () {
    int altezzacm;
    int altezzam;

    printf("Quanto sei alto? ");
    scanf("%d",&altezzacm);
    altezzam = altezzacm/100;
    printf("Sei alto %d m\n",altezzam);

    return 0;
}
```



```
Terminal
File Edit View Terminal Go Help
infob@infoB:/mnt/Lab2$ ./exe
Quanto sei alto? 186
Sei alto 1 m
infob@infoB:/mnt/Lab2$
```

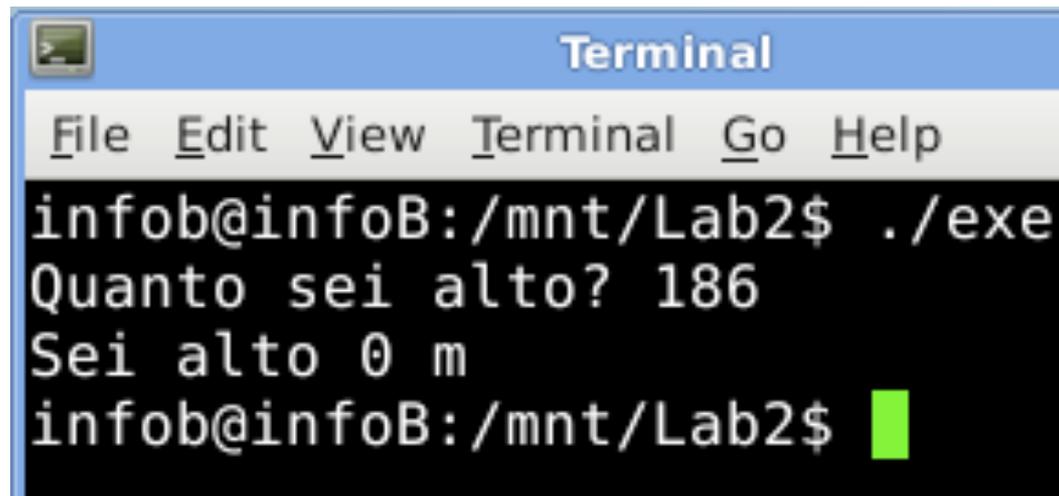
Un secondo errore

```
#include <stdio.h>

int main () {
    int altezzacm;
    float altezzam;

    printf("Quanto sei alto? ");
    scanf("%d",&altezzacm);
    altezzam = altezzacm/100;
    printf("Sei alto %d m\n",altezzam);

    return 0;
}
```



```
Terminal
File Edit View Terminal Go Help
infob@infoB:/mnt/Lab2$ ./exe
Quanto sei alto? 186
Sei alto 0 m
infob@infoB:/mnt/Lab2$
```

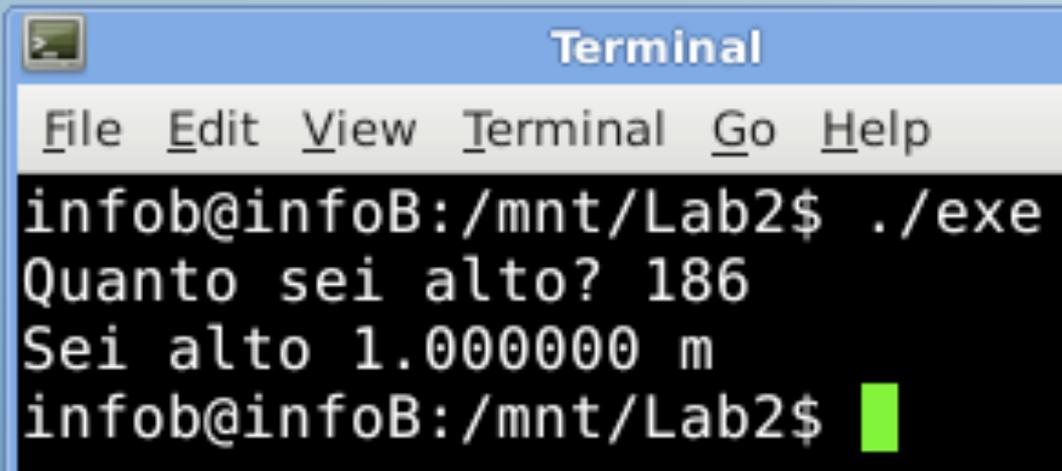
Un terzo errore

```
#include <stdio.h>

int main () {
    int altezzacm;
    float altezzam;

    printf("Quanto sei alto? ");
    scanf("%d",&altezzacm);
    altezzam = altezzacm/100;
    printf("Sei alto %f m\n",altezzam);

    return 0;
}
```



Terminal

```
File Edit View Terminal Go Help
infob@infoB:/mnt/Lab2$ ./exe
Quanto sei alto? 186
Sei alto 1.000000 m
infob@infoB:/mnt/Lab2$
```

➤ Soluzione corretta

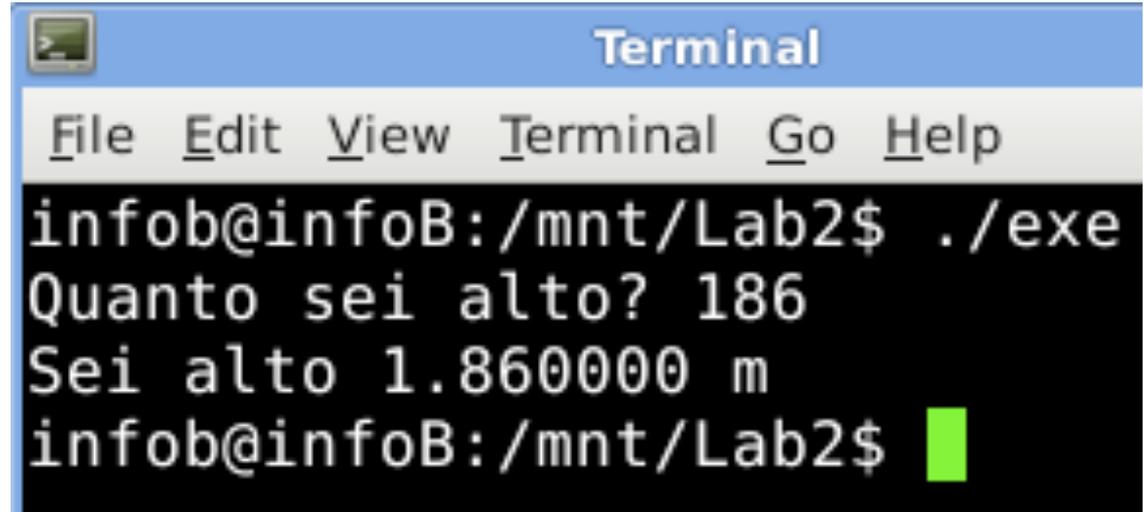
```
#include <stdio.h>
```

```
int main () {  
    int altezzacm;  
    float altezzam;  
    float rapporto = 100;
```

```
    printf("Quanto sei alto? ");  
    scanf("%d",&altezzacm);  
    altezzam = altezzacm/rapporto;  
    printf("Sei alto %f m\n",altezzam);
```

```
    return 0;
```

```
}
```



A terminal window titled "Terminal" with a menu bar containing "File", "Edit", "View", "Terminal", "Go", and "Help". The terminal output shows the program being executed in the directory "/mnt/Lab2". The prompt is "infob@infoB:/mnt/Lab2\$". The user enters ". /exe", and the program outputs "Quanto sei alto? 186" followed by "Sei alto 1.860000 m". The prompt returns to "infob@infoB:/mnt/Lab2\$".

L'importanza dei tipi di dato



- In C esistono diversi tipi di dato built-in, tra cui
 - int: numeri interi
 - float: numeri con virgola (singola precisione)
 - double: numeri con virgola (doppia precisione)
 - char: caratteri (sono interi che possono variare tra 0–255)
- Inoltre il C fornisce anche la possibilità di definire dei nuovi tipi di dato



Tipo carattere e codifica ASCII



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

```
int main () {
```

```
    char carattere;
```

```
    printf("Inserisci un carattere: ");
```

```
    scanf("%c",&carattere);
```

```
    printf("Il carattere inserito e': %c\n",carattere);
```

```
    printf("... ma anche: %d\n",carattere);
```

```
    return 0;
```

```
}
```

```
Terminal
File Edit View Terminal Go Help
infob@infoB:/mnt/Lab2$ ./exe
Inserisci un carattere: d
Il carattere inserito e': d
... ma anche: 100
infob@infoB:/mnt/Lab2$
```

WAT.

TUTTO È *NUMERI*
IL CALCOLATORE RAGIONA
SU QUESTI DATI





Tipo carattere e codifica ASCII



000	NUL	033	!	066	B	099	c	132	ä	165	Ñ	198	ã	231	þ
001	Start Of Header(SOH)	034	"	067	C	100	d	133	å	166	ª	199	Ä	232	Ë
002	Start Of Text (STX)	035	#	068	D	101	e	134	ä	167	º	200	Å	233	Û
003	End Of Text (ETX)	036	\$	069	E	102	f	135	ç	168	¿	201	Æ	234	Ü
004	End Of Transmission (EOT)	037	%	070	F	103	g	136	è	169	®	202	⌚	235	Ù
005	Enquiry	038	&	071	G	104	h	137	ë	170	¬	203	⌚	236	Ý
006	Acknowledge (ACK)	039		072	H	105	i	138	è	171	½	204	⌚	237	Ý
007	Bell	040	(073	I	106	j	139	ï	172	¼	205	=	238	—
008	Backspace (BS)	041)	074	J	107	k	140	î	173	ı	206	⌚	239	˘
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	«	207	⌚	240	-
010	Line Feed (LF)	043	+	076	L	109	m	142	Ä	175	»	208	ø	241	±
011	Vertical Tab	044	,	077	M	110	n	143	Å	176	⌚	209	Ð	242	_
012	Form Feed (FF)	045	-	078	N	111	o	144	É	177	⌚	210	Ê	243	¾
013	Carriage Return (CR)	046	.	079	O	112	p	145	æ	178	⌚	211	Ë	244	†
014	Shift Out	047	/	080	P	113	q	146	Æ	179		212	È	245	§
015	Shift In	048	0	081	Q	114	r	147	ö	180	†	213	ı	246	÷
016	Dataline Escape (DLE)	049	1	082	R	115	s	148	ö	181	À	214	í	247	˙
017	DC 1 (XON)	050	2	083	S	116	t	149	ò	182	Á	215	î	248	ª
018	DC 2	051	3	084	T	117	u	150	û	183	Â	216	ï	249	˚
019	DC 3 (XOFF)	052	4	085	U	118	v	151	ù	184	⊙	217	ĵ	250	¸
020	DC 4	053	5	086	V	119	w	152	ÿ	185	⌚	218	ŕ	251	˘
021	Negative Acknowledge (NAK)	054	6	087	W	120	x	153	Ö	186		219	■	252	˙
022	Synchronous Idle	055	7	088	X	121	y	154	Û	187	⌚	220	■	253	˚
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	⌚	221	ı	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	φ	222	ı	255	
025	End Of Medium	058	:	091	[124		157	∅	190	¥	223	■		
026	Substitute	059	;	092	\	125	}	158	×	191	γ	224	Ó		
027	Escape (ESC)	060	<	093]	126	~	159	f	192	ℒ	225	ß		
028	File Separator	061	=	094	^	127 (DEL)	␣	160	á	193	⊥	226	Ô		
029	Group Separator	062	>	095	_	128	Ç	161	í	194	⌚	227	Ö		
030	Record Separator	063	?	096	`	129	ü	162	ó	195	†	228	ø		
031	Unit Separator	064	@	097	a	130	é	163	ú	196	-	229	Ó		
032	SPACE (SP)	065	A	098	b	131	â	164	ñ	197	+	230	μ		



Più da vicino...

068	D	101	e
069	E	102	f
070	F	103	g
071	G	104	h
072	H	105	i

- Scrivere un programma che, letto un carattere minuscolo, lo riporta in maiuscolo
- Osservazione
 - Proviamo a stampare il valore ASCII di un carattere minuscolo e del suo corrispettivo maiuscolo



Maiuscolo: codice



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

int32_t main () {
    uint8_t c1, c2;

    printf("Inserisci un carattere minuscolo: ");
    scanf("%c",&c1);

    printf("Inserisci lo stesso carattere, ma in maiuscolo: ");
    scanf(" %c",&c2);

    printf("Il carattere inserito e' %c e in ASCII vale %u\n", c1, c1);
    printf("La sua rappresentazione maiuscola e' %c e in ASCII vale %u\n\n", c2, c2);

    return 0;
}
```



Maiuscolo: exe



```
Marcos-MacBook-Air:Desktop santa$ ./exe
Inserisci un carattere minuscolo: g
Inserisci lo stesso carattere, ma in maiuscolo: G
Il carattere inserito e' g e in ASCII vale 103
La sua rappresentazione maiuscola e' G e in ASCII vale 71
```

```
Marcos-MacBook-Air:Desktop santa$ ./exe
Inserisci un carattere minuscolo: a
Inserisci lo stesso carattere, ma in maiuscolo: A
Il carattere inserito e' a e in ASCII vale 97
La sua rappresentazione maiuscola e' A e in ASCII vale 65
```

```
Marcos-MacBook-Air:Desktop santa$ ./exe
Inserisci un carattere minuscolo: b
Inserisci lo stesso carattere, ma in maiuscolo: B
Il carattere inserito e' b e in ASCII vale 98
La sua rappresentazione maiuscola e' B e in ASCII vale 66
```

↘ Maiuscolo: -32!



Dato un carattere minuscolo, *dato*
Il corrispettivo maiuscolo è: *dato-32*