
Informatica Personale

Introduzione all'informatica

Dante Picca

Anno accademico 2009-2010

Che cos'è l'informatica

"Sono convinto che l'informatica abbia molto in comune con la fisica. Entrambe si occupano di come funziona il mondo a un livello abbastanza fondamentale. La differenza, naturalmente, è che mentre in fisica devi capire come è fatto il mondo, in informatica sei tu a crearlo. Dentro i confini del computer, sei tu il creatore. Controlli – almeno potenzialmente – tutto ciò che vi succede. Se sei bravo, puoi essere un dio. Su piccola scala."

Linus Torvalds, "Rivoluzionario per caso"
Editore Garzanti.

Che cos'è l'informatica

L'informatica [da *informat(ion) (automat)ique* Ph. Dreyfus 1962] è una scienza che studia i metodi e le tecniche per il trattamento automatico delle informazioni.

Per trattamento delle informazioni si intende l'acquisizione, la memorizzazione, la trasmissione ed il processamento.

Come sarà evidente tra poco, con l'avvento dei calcolatori elettronici, l'informatica si è concentrata sullo studio delle informazioni in formato binario, cioè all'informazione formata da un insieme di bit.

La conoscenza di **come** risolvere un problema e la **capacità** di risolverlo sono competenze distinte, da sempre possedute da entità diverse.

Nell'informatica, spetta **all'uomo** la competenza su come risolvere un problema, la **macchina** ha la capacità di farlo più velocemente e precisamente di noi!

Nel seguito:

Uomo e risolutore saranno sinonimi.

Computer o esecutore saranno sinonimi.

Le informazioni

Un elaboratore elettronico (ad esempio il vostro telefono cellulare), tratta informazioni in formato diverso, in particolare:

- numeri
- lettere
- suoni
- immagini.

Questo perché è stato definito un modo unico di rappresentare le diverse informazioni da parte dei sistemi informativi. In questo modo i numeri telefonici ed i suoni possono essere trattati in modo simile dai sistemi informativi.

Analogico e digitale

Si dice che una informazione è in formato **analogico**, quando il suo valore è ricavato per analogia (ad qui analogico) con un'altra grandezza fisica.

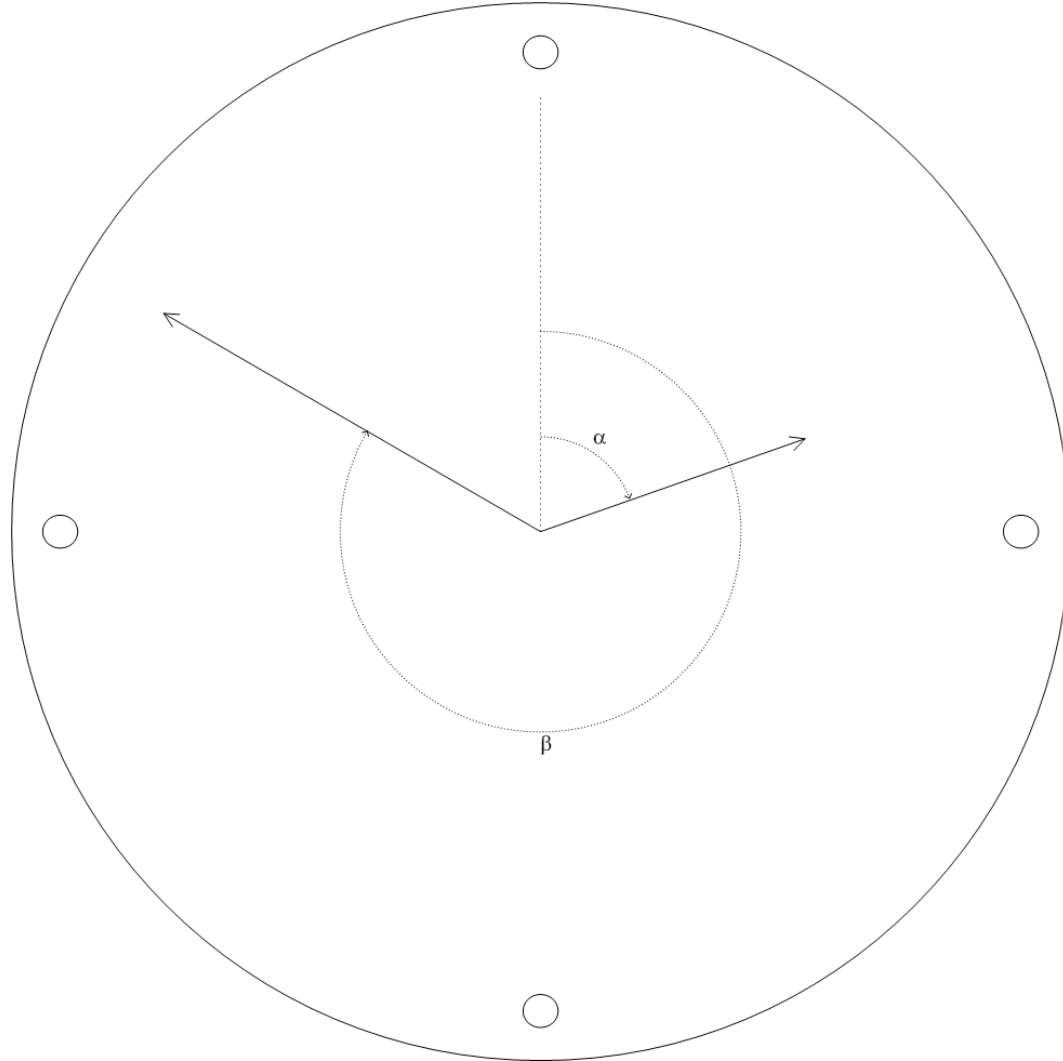
Si dice che una informazione è in formato **digitale**, quando il suo valore viene espresso sotto forma di numeri (in inglese digit, da cui digitale).

Esistono, intorno a noi, molti esempi di strumenti analogici o digitali per il trattamento delle informazioni. L'orologio è un esempio.

Analogico e digitale



Analogico e digitale



Il codice ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.asciitable.com

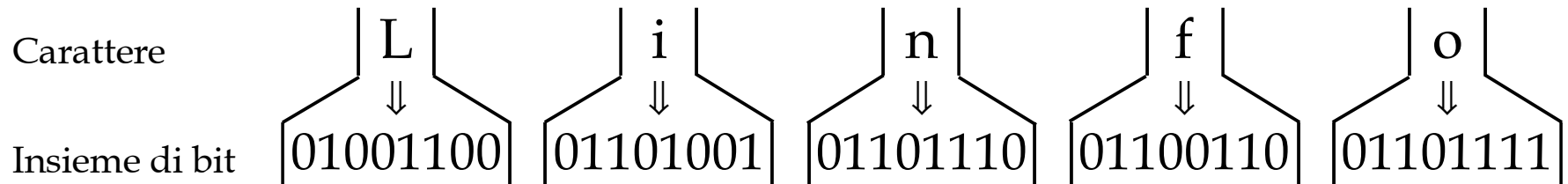
Che cos'è un bit

Un bit rappresenta l'elemento minimo d'informazione che può essere trattata da un calcolatore elettronico, così come un carattere è l'elemento minimo d'informazione che può essere trattato nel testo scritto.

A differenza di un carattere, che può assumere molti valori diversi (abcdefghijklmnopqrstuvwxyz,.- 0123456789 ...) un bit può assumere solo due valori: 0 ed 1 (od anche vero/falso oppure acceso/spento).

Che cos'è un bit

Tutte le informazioni trattate in forma binaria, sono composte da una sequenza ordinata di bit.



Che cos'è un bit

La numerazione binaria è uguale alla numerazione decimale, ma se un numero in numerazione decimale va da 0 a 9 (10 possibili valori), un numero in numerazione binaria vale 0 od 1 (2 possibilità). Possiamo sempre creare una tabella di corrispondenza tra le due numerazioni.

Decimale	Binario
0	0
1	1
2	10
3	11
4	100
5	101
10	1010
100	1100100
1.000	1111101000

Che cos'è un bit

Generalmente si utilizzano i multipli dei bit, poiché un singolo bit trasporta poche informazioni. I multipli più usati sono:

1 byte = 8 bit (da 0 a 255 in decimale)

1 word = 16 bit (da 0 a 65.535 in decimale)

1 Kbit = 1.024 bit

1 Mbit = 1.024 Kbit = 1.048.576 bit

1 Gbit = 1.024 Mbit = 1.073.741.824 bit

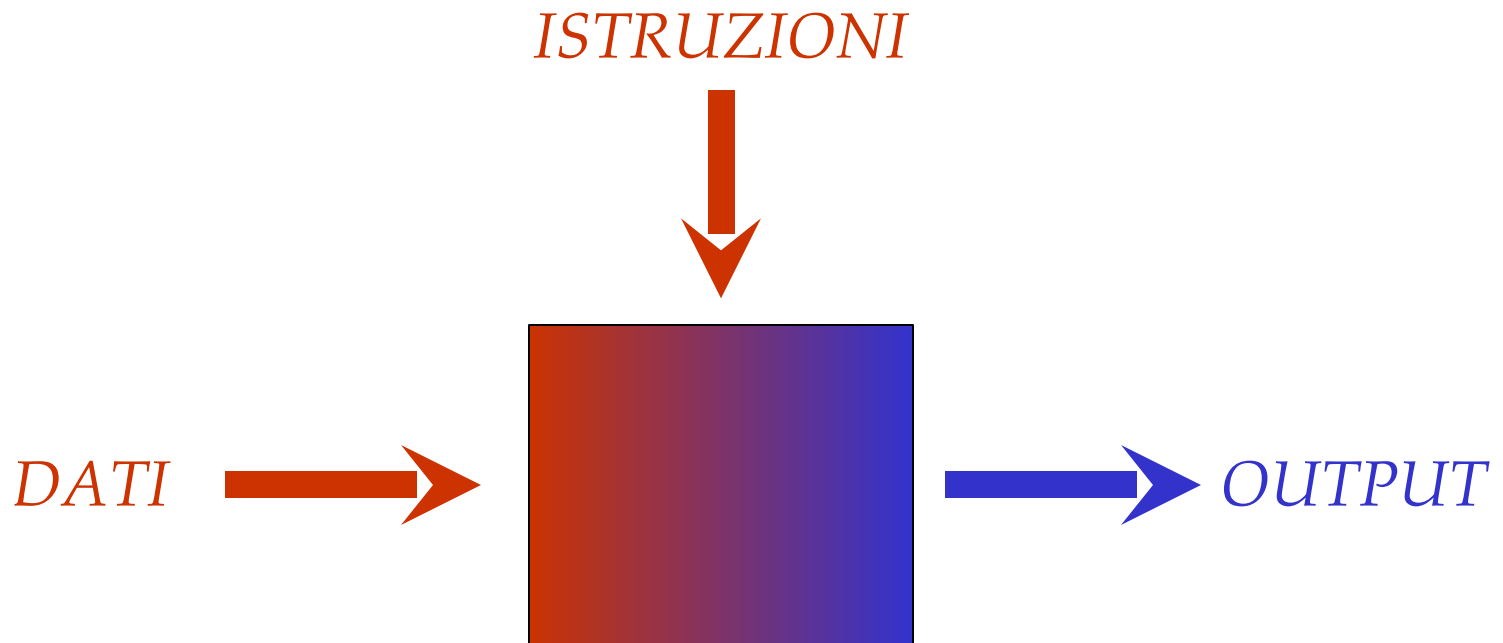
Che cos'è un computer

Un esecutore, ad esempio un computer, è una entità che produce dei dati in uscita (**output**) sulla base delle informazioni che riceve in ingresso (**input**).



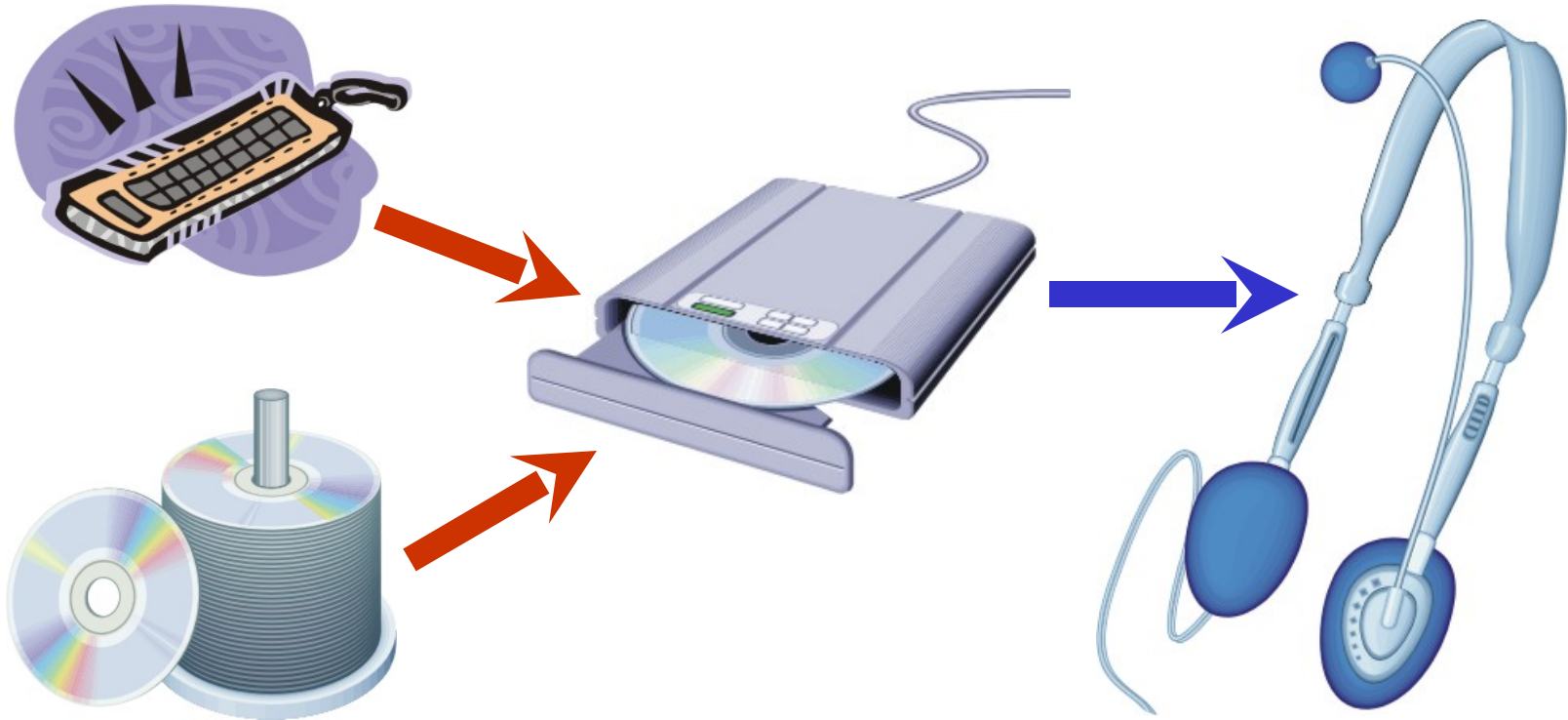
Che cos'è un computer

Per l'esecutore sono **input** tanto i **dati** sui quali deve lavorare, quanto le **istruzioni** che deve eseguire per trasformare i dati in **output**.



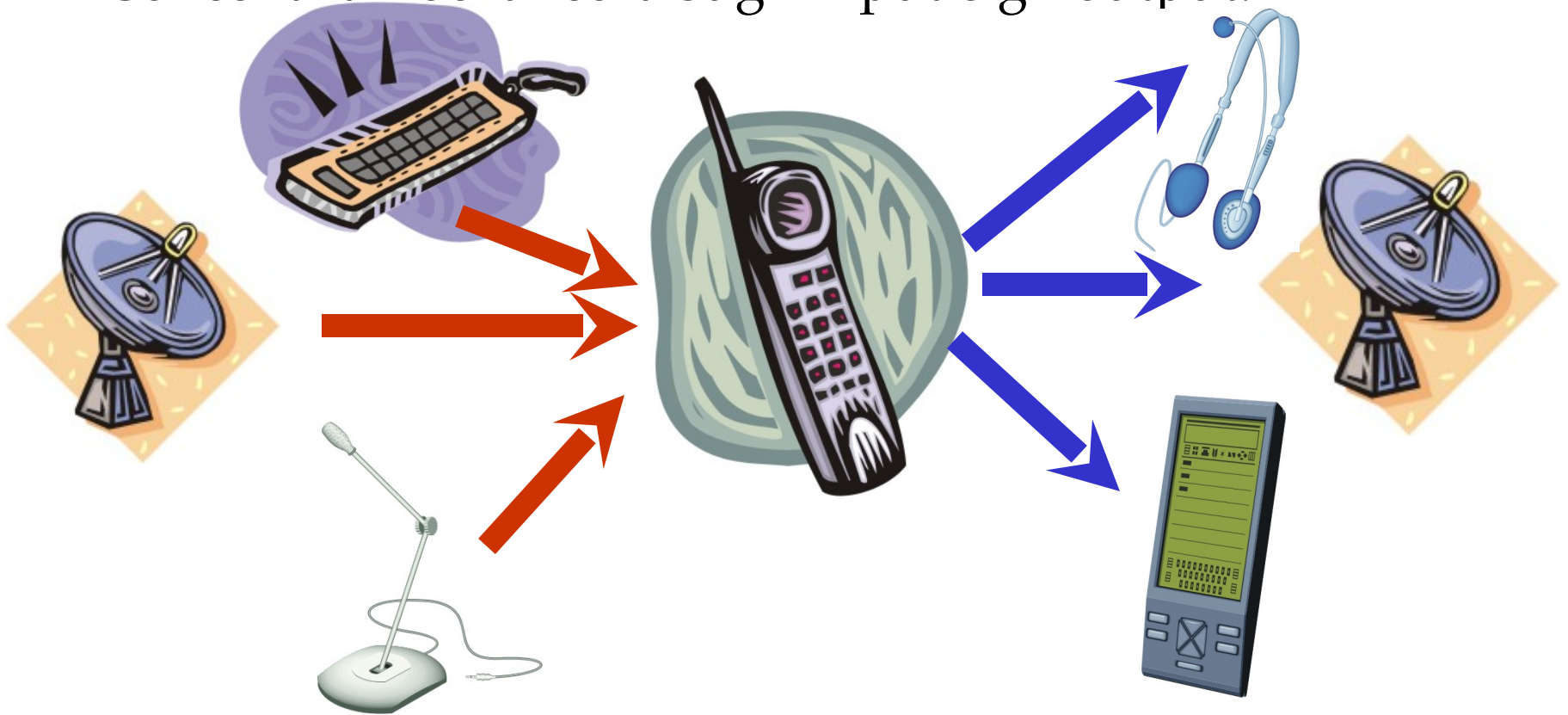
Che cos'è un computer

Un esempio di esecutore è il lettore di CD audio !
Concentriamoci sugli input e gli output.



Che cos'è un computer

Un altro esempio di esecutore è il Cellulare !
Concentriamoci ancora sugli input e gli output.



Che cos'è un computer

Nella definizione che abbiamo dato di esecutore rientra anche l'uomo !

Ognuno di noi, infatti, riceve degli input attraverso i sensi, li elabora attraverso il cervello e produce degli output, ad esempio parlando o scrivendo.

I calcolatori, tuttavia, sono solo uno strumento per esprimere la soluzione di un problema. Sono cioè dei rapidissimi **esecutori** di ordini forniti dall'uomo.

L'operazione di predisporre l'elenco degli ordini per **risolvere** un problema è una peculiarità dell'uomo !

Che cos'è un computer

Un computer è un rapidissimo esecutore di sequenze di istruzioni (gli algoritmi) fornite dall'uomo, mentre il nostro cervello elabora le informazioni grazie a milioni di anni di evoluzione Darwiniana.

Nel caso in cui un computer ed un uomo dovessero processare gli stessi input, sapremo esattamente secondo quali regole il computer ha prodotto degli output, mentre potremmo avere solo un'idea del processo seguito dall'uomo.

9 problemi che possono essere trattati da un elaboratore

Un problema è una questione da risolvere partendo da elementi noti, mediante il ragionamento.

I problemi si possono catalogare in tre macrocategorie:

1. problemi risolvibili
2. problemi non risolvibili
3. problemi non affrontabili

Un elaboratore può trattare solo la prima tipologia di problemi.

La risoluzione dei problemi

In generale, perché il risolutore possa descrivere all'esecutore la procedura da seguire per giungere alla soluzione di un problema, tale problema si scompone in sottoproblemi, eventualmente ancora scomponibili.

La scomposizione si ferma quando il risolutore arriva a trattare problemi elementari (o primitivi), non ulteriormente scomponibili, per la cui soluzione l'esecutore deve svolgere una o più azioni elementari.

Risolvere un problema equivale, così, ad **eseguire** una opportuna successione di problemi più semplici: le azioni elementari.

Questa tecnica di soluzione è detta TOP-DOWN.

Che cos'è un algoritmo

L'elaboratore elettronico per risolvere un problema utilizza un algoritmo, cioè un insieme di regole (o istruzioni) che, eseguite secondo un ordine prestabilito, permettono di trovare il risultato cercato sulla base dei dati in ingresso.

L'algoritmo è l'elencazione delle istruzioni da eseguire.

Nel seguito definiremo l'algoritmo come una sequenza di azioni che trasforma input in output attraverso un numero finito di passi elementari e non ambigui.

Il generico esecutore

La sequenza di azioni è valida solo per un insieme definito di dati iniziali, e può essere eseguita da un opportuno esecutore (che chiameremo “generico”), in grado tanto di comprendere ed eseguire le singole istruzioni che compongono i passi elementari quanto di operare sui dati in ingresso.

Non è necessario che l'esecutore conosca gli scopi o il senso complessivo dell'algoritmo, e per tale motivo è definito generico: un computer può quindi essere un generico esecutore di algoritmi.

L'attività di redazione dell'algoritmo da parte del risolutore viene svolta durante la terza fase del processo precedentemente descritto.

Che cos'è un algoritmo

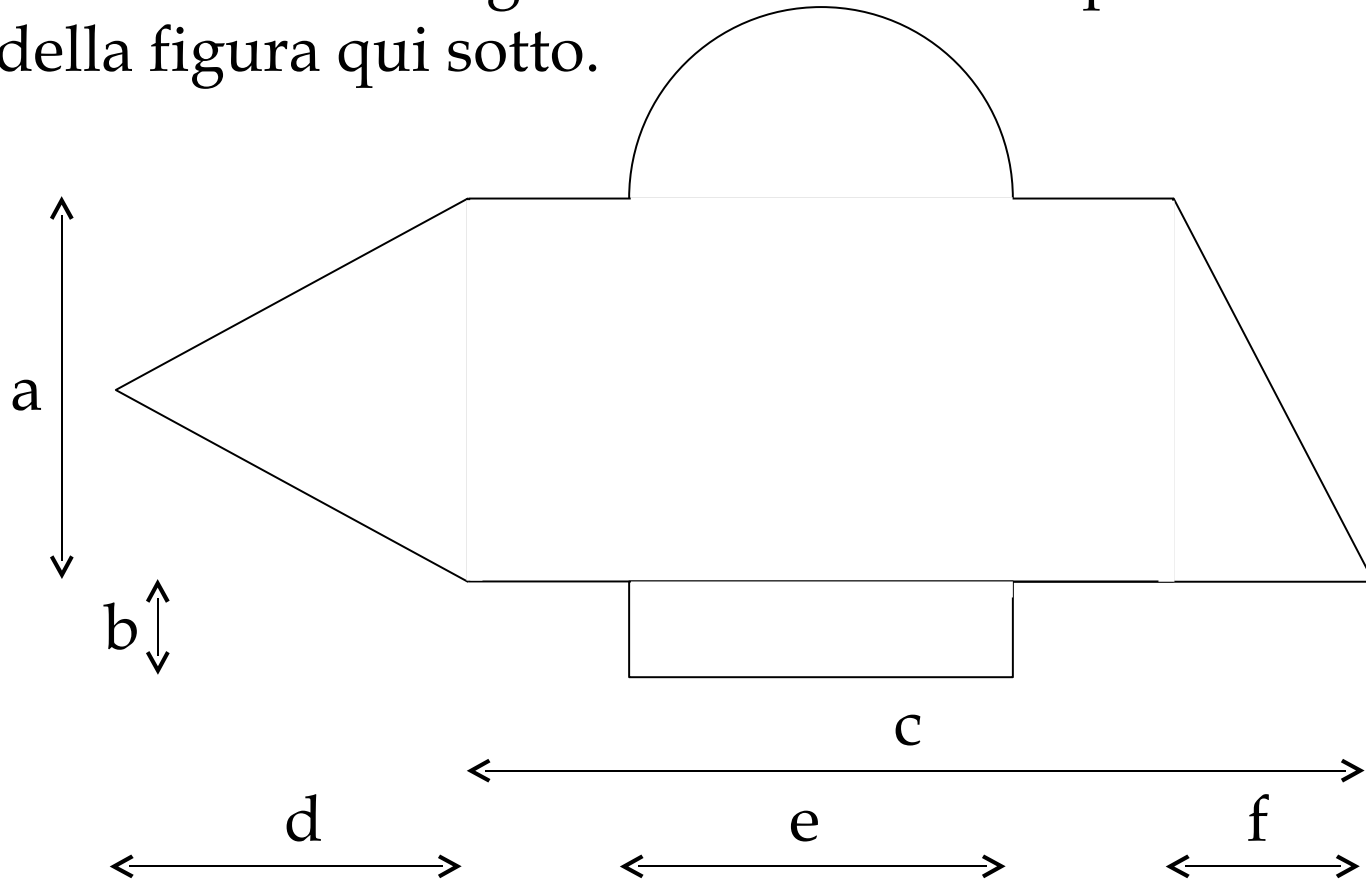
Gli algoritmi sono necessari in informatica: possono essere risolti autonomamente da un calcolatore elettronico senza bisogno dell'uomo.

Infatti possono essere eseguiti senza che l'esecutore sia a conoscenza degli scopi: anche un calcolatore può essere un valido esecutore!!

Il vantaggio nell'usare i computer risiede nella velocità di esecuzione e nella capacità di memorizzazione: un computer è in grado di calcolare il prodotto di due numeri di 100 cifre in una frazione di secondo senza errori, mentre un uomo impiegherebbe giorni ed il risultato potrebbe essere errato !

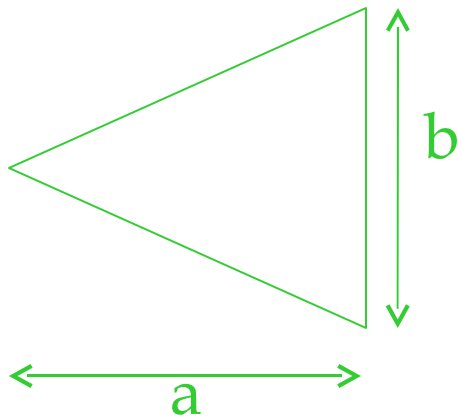
Un esempio di algoritmo

Introduciamo l'algoritmo con un esempio: il calcolo dell'area della figura qui sotto.

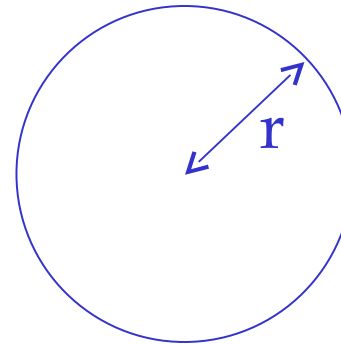


Un esempio di algoritmo

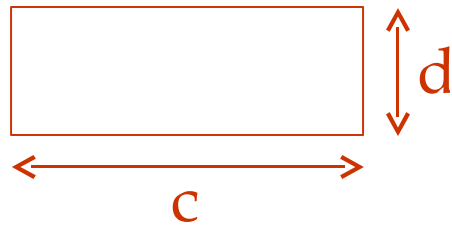
Noi non sappiamo calcolare l'area della figura, però sappiamo calcolare l'area di alcune figure elementari.



$$\text{Area}_T = \frac{a*b}{2}$$



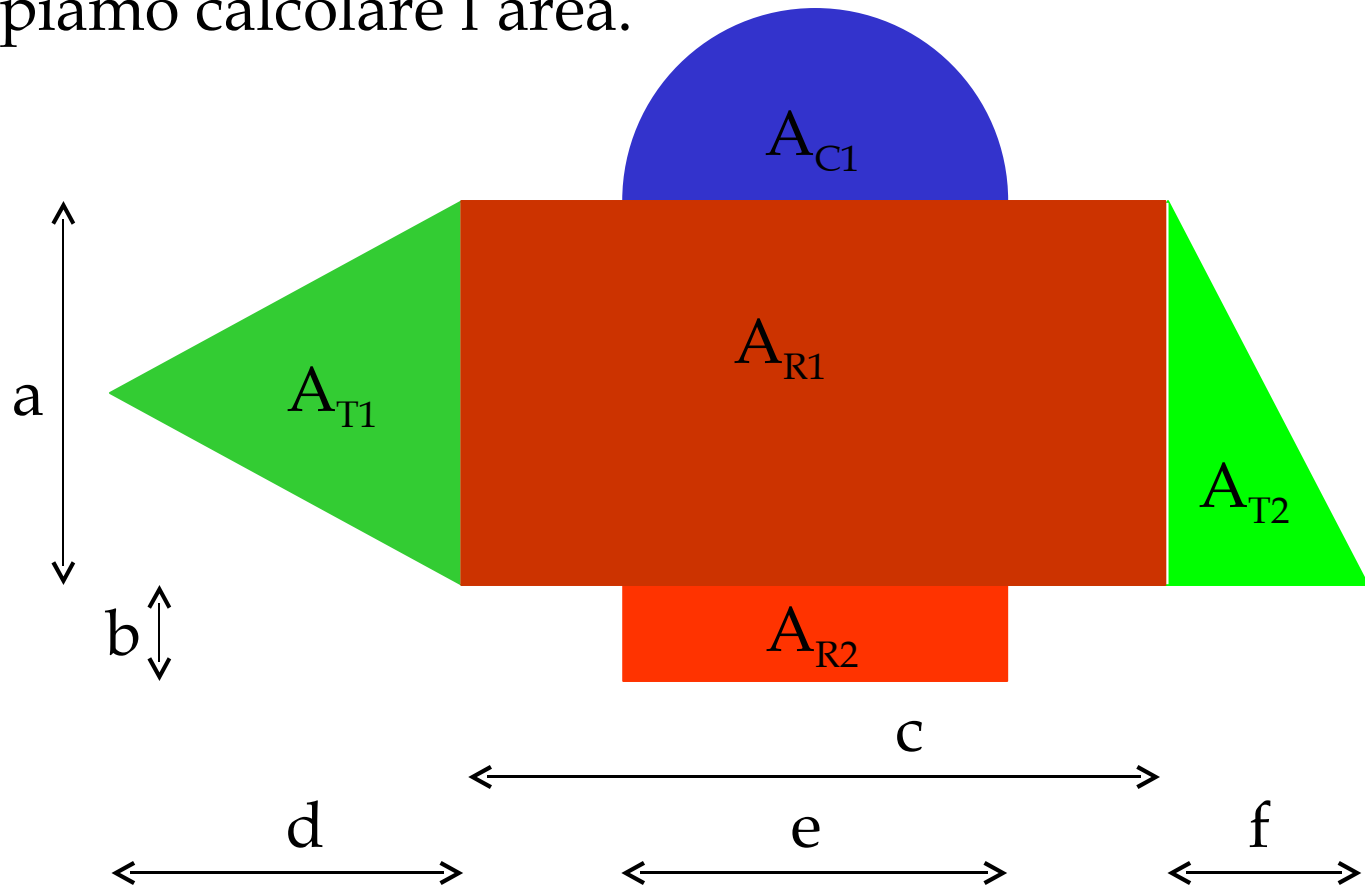
$$\text{Area}_C = \pi * r^2$$



$$\text{Area}_R = a*b$$

Un esempio di algoritmo

Possiamo separare la figura iniziale in figure elementari di cui sappiamo calcolare l'area.



Un esempio di algoritmo

Calcolare le aree delle diverse figure elementari è semplice:

$$\text{Area}_{T1} = \frac{a*d}{2}$$

$$\text{Area}_{R1} = a*c$$

$$\text{Area}_{C1} = \frac{\pi*(e/2)^2}{2}$$

$$\text{Area}_{T2} = \frac{a*f}{2}$$

$$\text{Area}_{R2} = b*e$$

L'area cercata risulta quindi

$$\text{Area}_{\text{FUGURA}} = \text{Area}_{T1} + \text{Area}_{T2} + \text{Area}_{R1} + \text{Area}_{R2} + \text{Area}_{C1}$$

Un esempio di algoritmo

Ricapitolando, per risolvere il problema:

- lo abbiamo scomposto in una serie di **azioni elementari** che sapevamo svolgere con facilità,
- abbiamo risolto le azioni elementari,
- abbiamo aggregato i risultati delle diverse azioni per produrre il risultato finale.

Dopo aver introdotto il concetto di algoritmo con un esempio, vediamo le sue proprietà e quelle delle singole azioni elementari.

Proprietà di un algoritmo

Una sequenza di istruzioni per essere definita algoritmo deve soddisfare 5 requisiti:

1. **Univocità**: non deve esistere alcuna libertà di azione od interpretazione per l'esecutore
 2. **Effettività**: le operazioni devono essere concluse in un tempo finito
 3. **Ingresso**: l'algoritmo deve operare su dati in ingresso
 4. **Uscita**: l'algoritmo deve produrre dati in uscita
 5. **Terminazione**: l'algoritmo deve terminare in un numero finito di passi elementari
-

Proprietà di un algoritmo

Anche le azioni elementari hanno delle proprietà, che discendono dalle proprietà dell'algoritmo, in particolare:

1. **Finitezza**: l'azione deve concludersi in un tempo finito, altrimenti si viola l'effettività
2. **Osservabilità**: ogni azione deve avere un effetto osservabile, altrimenti è inutile
3. **Riproducibilità**: ogni azione deve, a parità di input, produrre i medesimi output, poiché altrimenti si avrebbe una libertà da parte dell'esecutore: i risultati dell'azione elementare, quindi, devono essere riproducibili.

un esempio di algoritmo

Proviamo ora a costruire un algoritmo che svolga una moltiplicazione tra due numeri interi, come ad esempio $5*8$ oppure $132*439$.

Per far questo, supponiamo che il generico esecutore sappia solo fare le somme.

Sappiamo che moltiplicare due numeri n ed m , equivale a sommare m volte n :

$$8*5=8+8+8+8+8=40$$

un esempio di algoritmo

L'algoritmo, quindi, potrebbe essere:

- prendi in input due numeri **a** e **b**
- controlla che **a** e **b** siano interi, altrimenti scrivi in output che i numeri non vanno bene ed esci
- assegna alla variabile **somma** il valore 0
- ripeti **a** volte la seguente espressione: **somma=somma+b**
- scrivi in output il valore di **somma** ed esci

Esempio di algoritmo

Questa è la ricetta della parmigiana, secondo voi è un algoritmo ?

4 melanzane grosse, olio per friggere e farina q. b., 500 gr di passata di pomodoro, mezza cipolla, un ciuffo di basilico, tre cucchiaini d'olio d'oliva, 125 gr di parmigiano grattugiato, 250 gr di mozzarella tagliata a fettine, sale e pepe q. b.

Tagliare le melanzane a rondelle cospargerle con un pizzico di sale e lasciarle spurgare per un'ora. Asciugarle bene, infarinarle, e friggerle in abbondante olio caldo. Sgocciolarle bene con carta da cucina. Intanto preparare una salsa con l'olio d'oliva, la cipolla, il pomodoro, ed il basilico. In una teglia adagiate strati di melanzane, sugo, parmigiano e mozzarella. Infornare a 180°C per 30 minuti circa.

Esempio di algoritmo

Vediamo se la ricetta verifica le proprietà dell'algoritmo:

1. **Univocità**: per verificarlo devono essere analizzate le singole azioni elementari
2. **Effettività**: sicuramente sì, ci vorrà un paio d'ore
3. **Ingresso**: gli ingredienti
4. **Uscita**: la parmigiana
5. **Terminazione**: il numero di passi elementari è 7

Esempio di algoritmo

Cominciamo a separare le azioni elementari:

1. reperire gli **ingredienti**.
2. tagliare le melanzane a **rondelle**, cospargerle con un **pizzico** di sale e lasciarle spurgare per un'ora.
3. asciugarle **bene**, **infarinarle**, e **friggerle** in **abbondante** olio **caldo**.
4. sgocciolarle **bene** con carta da cucina.
5. **preparare una salsa** con l'olio d'oliva, la cipolla, il pomodoro, ed il basilico.
6. adagiate in una teglia **strati** di melanzane, sugo, parmigiano e mozzarella.
7. infornare a 180°C per 30 minuti **circa**.

Esempio di algoritmo

Ricapitolando, tutte le azioni verificano le proprietà di **finitezza** ed **osservabilità**, ma molte non verificano la proprietà della **riproducibilità**.

Inoltre alcune azioni richiamano altri algoritmi (ad esempio **friggere** o **preparare una salsa**) che dovrebbero essere analizzati, perché a priori potrebbero non verificare tutte le proprietà !

Per rendere la ricetta un algoritmo, dovremmo modificare il testo, ad esempio:

tagliare le melanzane a rondelle, cospargerle con un pizzico di sale diventerebbe:

tagliare le melanzane a rondelle spessore $(0,75 \pm 0.05)$ cm e cospargere ogni rondella con 1.5 grammi di sale ...

Il generico esecutore

Una volta compreso l'algoritmo, possiamo immaginare quali debbano essere le proprietà di un generico esecutore, cioè dell'elaboratore che dovrà eseguire tutti i passaggi per trasformare gli input in output seguendo le azioni elementari.

Lo schema che seguiremo è detto **modello della macchina di Von Neumann** ed è una descrizione **funzionale/strutturale**, cioè racchiude le caratteristiche essenziali di un generico calcolatore, senza entrare nel merito dei singoli aspetti implementativi.

La macchina di Von Neumann è il paradigma seguito oggi nella progettazione dei calcolatori elettronici.

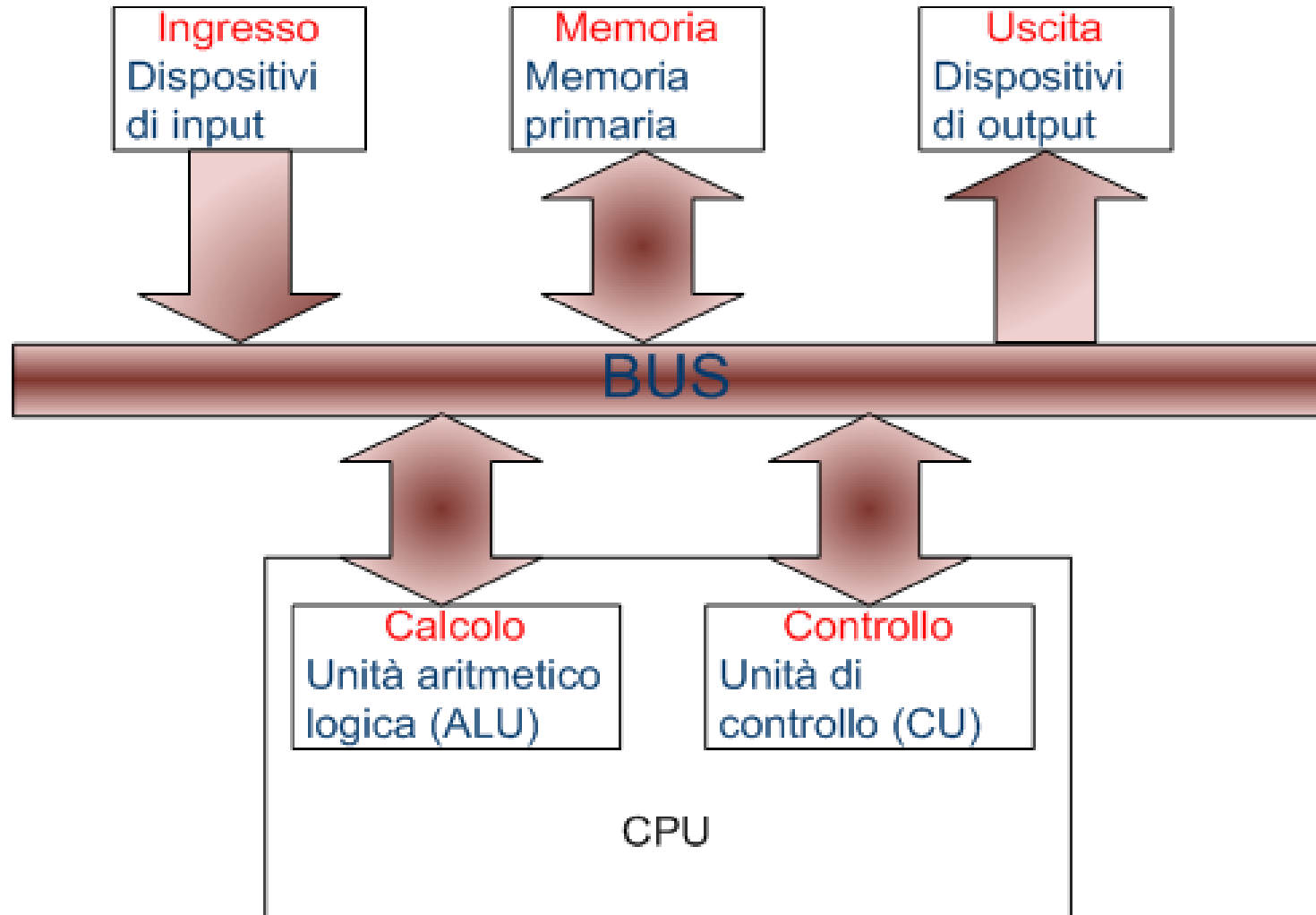
La macchina di Von Neumann

Quali funzioni dovrà svolgere il generico esecutore?

1. accettare i dati in ingresso (input)
2. produrre i dati in uscita (output)
3. memorizzare i dati (iniziali ed intermedi) e le istruzioni
4. svolgere le operazioni matematiche necessarie per l'esecuzione
5. dirigere e controllare le precedenti funzioni al fine di eseguire l'algoritmo

I componenti che svolgono le diverse funzioni devono essere collegati tra di loro (ad esempio i dati in ingresso devono arrivare alla memoria), per questo sono connessi attraverso un **BUS**.

La macchina di Von Neumann



La macchina di Von Neumann: dispositivi di I/O

I dispositivi di ingresso ed uscita (I/O) svolgono la funzione di acquisire i dati e le istruzioni su cui l'esecutore deve operare: sono l'interfaccia tra la macchina ed il mondo esterno !



La macchina di Von Neumann: la memoria primaria

La memoria primaria svolge la funzione di memorizzare temporaneamente:

- le istruzioni sulle quali operare
- i dati acquisiti inizialmente e quelli che via via l'esecutore ha la necessità di memorizzare per portare a termine il suo compito.

Nell'esempio del calcolo dell'area, i dati iniziali sono i valori dei lati (a,b,c,d,e,f) ed i dati temporanei sono le aree ottenute dal calcolo delle figure elementare e che vengono sommate prima di restituire, in uscita, l'area della figura.

$$\text{Area}_{\text{FUGURA}} = \text{Area}_{\text{T1}} + \text{Area}_{\text{T2}} + \text{Area}_{\text{R1}} + \text{Area}_{\text{R2}} + \text{Area}_{\text{C1}}$$

La macchina di Von Neumann: l'unità aritmetico logica

L'unità aritmetico logica (**ALU**) svolge la funzione di calcolo sui dati. A questa unità è quindi demandato il compito di operare sui dati, ad esempio con somme, sottrazioni od operazioni logiche di confronto.

Nell'esempio del calcolo dell'area, è compito della ALU eseguire le moltiplicazioni, le divisioni e le somme.

La macchina di Von Neumann: l'unità di controllo

L'unità di controllo (**UC**) svolge la funzione di coordinamento sulle unità precedentemente descritte. Alcuni compiti di questa unità sono:

- determinare le operazioni di trasferimento dati
- indicare alla ALU le operazioni da svolgere nonché i relativi dati
- decidere quale istruzioni eseguire sulla base delle risposte ricevute dalla ALU.

Nei computer le funzioni della ALU e della UC sono integrate in un unico dispositivo fisico: la **CPU**, detta anche **il processore**

La macchina di Von Neumann: il bus

Il bus è un dispositivo fisico che consente lo scambio di dati ed istruzioni tra le diverse unità.

È una sorta di autostrada che permette, ad esempio, al processore di salvare nella memoria primaria i dati ricevuti dal dispositivo di input.

Alcuni dispositivi possono accedere al bus solo per scrivere dei dati (ad esempio i dispositivi di input), altri solo per leggere (ad esempio i dispositivi di output), altri sia per scrivere che per leggere (ad esempio la memoria primaria).

La memoria

Esistono due tipologie di memorie utilizzate nei calcolatori elettronici:

- **memoria primaria o principale**: dispositivi veloci ($\sim 1 \text{ GB/s}$) di memorizzazione volatile, con capienza limitata ed alto costo per MB memorizzato (RAM, cache)
- **memoria secondaria o di massa**: dispositivi lenti ($\sim 1 \text{ MB/s}$) di memorizzazione permanente, con elevate dimensioni e bassi costi per MB memorizzato (Hard disk, CD-Rom). Per Von Neumann questi sono dei dispositivi di I/O

La memoria

Prima distinzione tra le tipologie di memoria.

- permanente** se mantiene le informazioni che contiene anche in assenza di alimentazione, come accade per CD, HD e videocassetta.
- volatile** o **temporanea** se perde le informazioni che contiene quando viene a mancare l'alimentazione, come accade per la RAM dei computer, o per gli orologi digitali che perdono l'ora.

La memoria

Seconda distinzione tra le memorie.

- **memorie ad accesso casuale** [arbitrario] (Random Access Memory - **RAM**) consentono l'accesso in lettura e scrittura alle informazioni in modo diretto, attraverso un indirizzo. Nei CD audio è possibile ascoltare una canzone senza prima aver ascoltato le precedenti.
- **memorie di sola lettura** (Read Only Memory - **ROM**), sono come le RAM, ma una volta inserite le informazioni, queste non sono più cancellabili.
- **memorie ad accesso sequenziale** (Sequential Memory - **SM**) l'accesso alle informazioni è sequenziale, come avviene per le scene di un film presenti in una videocassetta.

I costituenti dei PC

Vediamo ora alcuni componenti presenti nei PC.

Case: è il contenitore esterno del PC

Motherboard (o scheda madre): è una scheda sulla quale sono presenti i bus e dove vengono collegati il processore, la RAM e gli altri dispositivi interni, è interna al case.

Processore: è la CPU, ossia il "cervello" del computer. I principali produttori sono AMD ed INTEL. Ciò che caratterizza i processori è la velocità di esecuzione, espressa in Hertz (HZ) o suoi multipli. I processori attuali hanno velocità intorno ai 3-4 GHz. È interno al case.

I costituenti del PC

RAM: è la memoria principale del computer. La quantità di informazioni tipicamente memorizzabili nella RAM presente nei PC è di 3-4 GB. È interna al case.

Hard disk: è la memoria secondaria e permanente utilizzata dal computer per memorizzare dati e programmi. La quantità di informazioni tipicamente memorizzabili è di 300-500 GB per hard disk. È interno al case.

Lettore di floppy-disk: consente di far leggere al computer dischi rimovibili di piccole dimensioni. Ormai stanno scomparendo perché poco affidabili. È interno al case, ma sul case è presente una fessura per inserire i dischi.

Lettore di CD o DVD: consente di far leggere al computer CD e DVD. Per pochi euro in più è possibile acquistare anche il masterizzatore, che consente di scrivere CD-R e DVD-R vergini. È interno al case, ma sul case è presente una fessura per inserire i dischi.

Scheda audio: consente al computer di inviare suoni alle casse e di ricevere suoni da un microfono. Il più noto produttore è la SoundBlaster. È interna al case, ma sul case sono presenti i connettori per le casse ed il microfono.

Scheda di rete: consente al computer di inviare e ricevere informazioni attraverso una rete.

Scheda video: consente al computer di inviare immagini al monitor. È interna al case, ma sul case è presente il connettore per il monitor.

Porte seriali, parallele, PS2 od USB: sono dei connettori presenti sul case e collegati alla scheda madre, che consentono a dispositivi esterni (stampante, mouse, tastiera, modem) di collegarsi al computer.

Modem: consente al computer di collegarsi ad altri computer (ad esempio ad internet) utilizzando la linea telefonica. Può essere interno od esterno al case. In quest'ultimo caso si collega al computer utilizzando la porta seriale od USB.

Hardware e Software

HARDWARE: componenti elettronici e meccanici del computer e delle periferiche (i dispositivi della macchina di Von Neumann).

SOFTWARE: l'insieme dei programmi necessari al funzionamento del computer (software di sistema) od alla soluzione di specifici problemi (software applicativo).

Le realizzazioni della macchina di Von Neumann sono l'hardware, la trasposizione degli algoritmi in un linguaggio comprensibile dal computer è il software