



# EIPASS Junior

Ei-Book | Scuola Primaria

## Premessa

La nascita, lo sviluppo e il consolidamento delle competenze digitali di base in soggetti in età scolare (8-13 anni) rappresentano, a tutt'oggi, una fra le sfide di maggiore interesse culturale, economico e sociale non solo del nostro Paese, ma di tutta la Comunità Europea.

Sin dalla pubblicazione della "Raccomandazione del Parlamento Europeo relativa a competenze chiave per l'apprendimento permanente" del 18 dicembre 2006, che in Italia ha dato origine alle "Competenze chiave per la cittadinanza attiva", Scuola e mondo della formazione hanno iniziato a formulare proposte concrete per impostare correttamente un rinnovato curriculum formativo nel settore ICT destinato alle fasce primarie dell'istruzione.

Il punto di partenza per il nuovo Programma "EIPASS® Junior" è costituito proprio dalle "Competenze per la cittadinanza attiva", con un particolare accento su:

- Imparare ad imparare
- Progettare
- Comunicare
- Collaborare e partecipare
- Agire in modo autonomo e responsabile
- Risolvere problemi
- Individuare collegamenti e relazioni
- Acquisire e interpretare l'informazione

Le competenze digitali assumono per il contesto europeo un carattere trasversale e quanto mai strategico; l'opportunità di utilizzare in modo appropriato ed efficace le Tecnologie dell'Informazione e della Comunicazione è oramai ritenuta requisito di base, e visto anche nell'ottica del corretto avvicinamento dell'Individuo alle forme di apprendimento formale, non formale e informale.

Il nuovo Programma "EIPASS® Junior" affonda le sue basi sugli elementi d'informatica che lo stesso Ministero indica come fattori essenziali per la crescita e lo sviluppo di competenze interdisciplinari nel giovane allievo, non a torto ritenuti fattori essenziali per la strutturazione di un solido curriculum formativo nell'Istruzione Primaria e Secondaria di Primo Grado.

Infatti, il Syllabus di Elementi di Informatica per la scuola dell'Obbligo (pubblicato dal MIUR nel 2010) specifica chiaramente quali siano gli obiettivi che giustificano l'introduzione dell'Informatica in quella fascia di riferimento:

- Iniziare ad utilizzare la logica nelle attività di organizzazione della conoscenza e nella costruzione delle competenze;
- Conoscere le basilari metodologie e tecniche della programmazione, dell'algorithmica e della rappresentazione dei dati, come risorsa concettuale utile ad acquisire e a saper usare competenze e abilità generali di problem solving;
- Possedere le abilità strumentali che consentono di usare i servizi offerti da Internet e dai software didattici, disponibili per ogni disciplina del curriculum.

Il tutto perfettamente inserito nel più ampio quadro di competenze, che il giovanissimo studente e cittadino deve iniziare a conoscere e padroneggiare, per poter godere dei benefici della "cittadinanza responsabile nella società dell'informazione".

Nel panorama attuale, definire il nuovo Programma "EIPASS® Junior" un semplice percorso di certificazione delle competenze ICT è decisamente riduttivo. Il suo ambizioso obiettivo è quello di affiancare ed integrare l'operato dei Docenti nel programmare e attuare percorsi per l'insegnamento della disciplina informatica e nello sviluppo di buone prassi relative all'utilizzo di programmi applicativi informatici.

Il Programma intende quindi accompagnare il Docente nella organizzazione del Piano dell'Offerta Formativa e della Programmazione didattica con la sistematizzazione di concetti, abilità e contenuti a carattere digitale, tali da includere non più e non soltanto l'uso corretto di un computer, ma la stimolazione di processi e strutture logiche di pensiero; solo in questo modo si potrà favorire la nascita e l'armonioso sviluppo di abilità e competenze proprie di quell'autonomia cognitiva che rappresenta l'obiettivo finale del processo di apprendimento.

*Certipass*

*Comitato Tecnico-Scientifico*

## Disclaimer

Certipass ha redatto il presente documento programmatico in base agli standard e ai riferimenti Comunitari vigenti in materia di competenze a carattere digitale. Certipass non si assume alcuna responsabilità derivante dall'applicazione in ambito diverso dallo stesso, neanche da informazioni elaborate da terzi in base ai contenuti del presente Programma.

Certipass si riserva di aggiornare il presente documento a propria discrezione, in ogni momento e senza darne preavviso, pubblicando le modifiche effettuate. L'Utenza destinataria è tenuta ad acquisire in merito periodiche informazioni visitando le aree del sito dedicate al Programma.

## Copyright

È vietata qualsiasi riproduzione, anche parziale, del presente documento senza preventiva autorizzazione scritta da parte di Certipass (Unico Ente Erogatore della Certificazione Informatica Europea EIPASS®). Le richieste di riproduzione devono essere inoltrate a Certipass.

Il logo EIPASS® è di proprietà esclusiva di Certipass. Tutti i diritti riservati.

## INDICE

<b>1. Elementi di base dell'informatica</b> .....	<b>7</b>
1.1 Il codice binario.....	7
1.1.1 Le basi di raggruppamento.....	7
1.1.2 La struttura binaria dell'informazione.....	7
1.1.3 Dal bit al byte.....	9
1.2 Approccio agli algoritmi e ai diagrammi di flusso.....	10
1.2.1 Concetto di algoritmo.....	10
1.2.2 Analisi di un semplice algoritmo.....	10
1.2.3 Concetto di diagramma di flusso (flow chart).....	11
1.3 Rappresentazione e risoluzione di semplici problemi.....	13
1.3.1 Concetto di diagramma.....	13
1.3.2 I diagrammi e l'informatica.....	14
1.3.3 Diagramma di risoluzione di un semplice problema.....	15
1.4 Classificazione delle informazioni.....	15
1.4.1 Definizione di grafico e di tabella.....	15
1.4.2 Rappresentare una serie di dati mediante grafici e tabelle.....	16
<b>2. Analisi delle componenti hardware di un computer</b> .....	<b>18</b>
2.1 L'hardware.....	18
2.1.1 Componenti principali di un computer.....	18
2.1.2 Unità Centrale di Elaborazione (CPU).....	19
2.1.3 Memoria.....	19
2.1.4 Dispositivi di memoria di massa.....	20
2.1.5 Archivi di tipo analogico e di tipo digitale.....	22
2.2 Classificazione dei computer.....	27
2.2.1 Le più comuni tipologie di computer.....	27
<b>3. Gestione di un sistema operativo a interfaccia grafica (elementi di base)</b> .....	<b>29</b>
3.1 Software.....	29
3.1.1 Software di sistema.....	29
3.1.2 Software applicativo.....	29
3.2 Sistema Operativo (GUI).....	29
3.2.1 Concetto di Sistema Operativo.....	29
3.2.2 Avviare il computer.....	30
3.2.3 Spegnerne il computer.....	30
3.2.4 Finestra di dialogo.....	31
3.2.5 Finestra applicativa.....	31
3.3 Uso delle icone.....	32
3.3.1 Le icone più comuni.....	32
3.3.2 Selezionare e spostare le icone sul desktop.....	32
3.4 Gestione menu e finestre.....	32
3.4.1 Definizione di menu.....	32

3.4.2	Operazioni di base su una finestra.....	33
<b>3.5</b>	<b>Gestione file e cartelle .....</b>	<b>34</b>
3.5.1	Concetto di file.....	34
3.5.2	Concetto di file.....	34
3.5.3	Concetto di cartella .....	34
3.5.4	Operazioni di base su file e cartelle.....	35
<b>4.</b>	<b>Fondamenti per la produttività informatica .....</b>	<b>36</b>
<b>4.1</b>	<b>Elaborare testi .....</b>	<b>36</b>
4.1.1	Primi passi con un semplice elaboratore di testi .....	36
4.1.2	La finestra applicativa e la barra degli strumenti.....	36
4.1.3	Operazioni di base su un documento di testo .....	38
<b>4.2</b>	<b>Elaborare una presentazione .....</b>	<b>38</b>
4.2.1	Primi passi con un semplice elaboratore di presentazioni .....	38
4.2.2	Operazioni di base su una presentazione.....	39
4.2.3	Operazioni di base su una presentazione.....	40
<b>4.3</b>	<b>Creare un disegno .....</b>	<b>41</b>
4.3.1	Operazioni di base su un disegno .....	41
<b>5.</b>	<b>Fondamenti di navigazione e ricerca di informazioni sul Web.....</b>	<b>43</b>
<b>5.1</b>	<b>Che cosa è il Web .....</b>	<b>43</b>
5.1.1	Concetto di rete e di Web .....	43
5.1.2	Definizione di browser .....	43
5.1.3	Operazioni di base su un browser.....	44
5.1.4	Utilizzare i segnalibri .....	45
5.1.5	Definizione di motore di ricerca .....	46
5.1.6	Effettuare semplici ricerche sul Web utilizzando le parole chiave.....	46

## 1. Elementi di base dell'informatica

### 1.1 Il codice binario

#### 1.1.1 Le basi di raggruppamento

Nella vita quotidiana ci troviamo continuamente a risolvere problemi matematici più o meno difficili. Per esempio quando andiamo al supermercato e dobbiamo pagare ciò che abbiamo scelto, facciamo dei conti in mente per dare al cassiere la banconota giusta ed aspettarci o meno il resto.

Per poter fare queste operazioni abbiamo bisogno di comunicare le "informazioni" in modo chiaro a chi ci ascolta. Nell'esempio precedente, noi ed il cassiere parliamo la stessa lingua e facciamo uso del "sistema metrico decimale" per calcolare il costo della spesa: cioè abbiamo "basi" comuni per comunicare.

Un numero può essere espresso secondo diversi modi di raggruppamento, denominati "basi". Una base indica, quindi, in che modo sono "raggruppate" le unità. Così, la "base dieci" sta ad indicare che, per rappresentare una determinata quantità, raggrupperemo le unità che la compongono a gruppi di dieci. Per esempio, il numero "12" sarà formato da:

- > due unità;
- > una decina (ovvero un gruppo di dieci unità).

Le due unità sono considerate da sole, in quanto non riescono a formare una ulteriore decina. La somma delle quantità rappresentate darà come risultato la quantità "dodici".

Il modo più semplice per raggruppare delle unità, ovvero degli oggetti, consiste nell'uso delle dita di una mano: ad ogni dito corrisponde un'unità e, al raggiungimento di un gruppo di dieci (tante unità quante sono le dita delle nostre mani) si mette da parte una decina.

Con il sistema di numerazione che usiamo di solito, quello decimale, possiamo combinare le 10 cifre da 0 a 9 per rappresentare tutti i numeri ed eseguire tutte le operazioni. In questo sistema la posizione di ciascuna cifra ha un preciso significato, per questo è definito "sistema posizionale".

Per esempio nel numero 346, la cifra 6 indica le unità, il 4 le decine e il 3 le centinaia.

La base dieci, però, non rappresenta l'unico modo matematico possibile di raggruppamento: cerchiamo di capire il sistema di numerazione binaria (base 2) di cui si serve il computer.

#### 1.1.2 La struttura binaria dell'informazione

Possiamo immaginare il computer formato da un grande numero di circuiti attraverso i quali può passare o meno la corrente elettrica: se indichiamo con il numero 1 il circuito chiuso che consente il passaggio della corrente e con il numero 0 il circuito aperto che blocca il passaggio della corrente, possiamo rappresentare diverse situazioni.

Per capire meglio, consideriamo quattro circuiti nei seguenti stati:

- > aperto chiuso chiuso aperto

che possiamo rappresentare così 1 0 0 1.

Il computer "conosce" soltanto i numeri "0" e "1", si basa cioè sul sistema numerico binario.

L'elemento base sul quale funziona il computer è il bit (binary digit = cifra binaria), cioè le cifre 0 e 1 nel sistema di numerazione binario. In questo sistema di numerazione la base è 2, cioè si effettuano raggruppamenti per 2.

Vediamo come un'informazione può essere espressa dal sistema binario.

Abbiamo invitato a casa nostra un amico, ma non siamo certi di esserci per l'ora dell'appuntamento. La situazione è questa:

- abitiamo all'ultimo piano di un palazzo di dodici piani;
- l'ascensore é guasto;
- il telefono non funziona.

Come possiamo fare per evitare che il nostro amico salga inutilmente?

Una soluzione potrebbe essere la seguente: diremo al nostro amico, una volta giunto sotto il palazzo in cui abitiamo, di guardare verso le nostre finestre. Se vede le luci accese potrà essere certo di trovarci in casa, se le luci sono spente, invece, significa che non siamo ancora in casa.

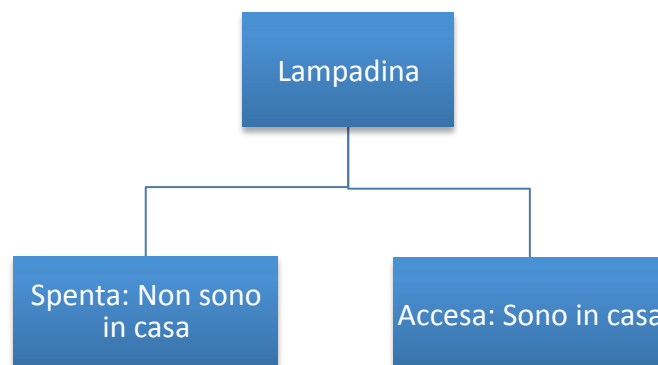
Questa situazione può riassumersi in questo modo:

- luci accese: siamo in casa
- luci spente: non siamo in casa

In base a questa informazione, il nostro amico si comporterà in due modi diversi:

- luci accese: il nostro amico salirà
- luci spente: il nostro amico andrà via

In pratica, sfruttando lo "stato" di una sola lampadina (accesa o spenta), siamo in grado di rappresentare 2 informazioni:



Sostituendo a ciascuno "stato" della lampadina un simbolo scelto fra quelli disponibili in codice binario ("0" e "1") avremo:

- 0 → non sono in casa
- 1 → sono in casa

Un codice è, generalmente, un insieme di simboli in grado di rappresentare un'idea, un concetto.

L'esempio della lampadina riassume il significato del codice binario:

- Il codice binario consente di rappresentare con i simboli "0" e "1" lo stato di "spento" o "acceso" di un interruttore; questo stato corrisponde ad una informazione.
- L'interruttore, che corrisponde ad un circuito digitale, prende il nome di bit (binary digit).
- Una memoria digitale è costituita da una serie di bit, ossia di interruttori che, a seconda del loro stato di "acceso" o "spento", sono in grado di rappresentare un numero molto elevato di informazioni.

- Una memoria digitale assume questo nome perché il termine "digit" ricorda lo stato di un dispositivo (acceso-spento), che viene modificato attraverso l'utilizzo delle dita (servono per premere tasti e interruttori, provocando uno dei due stati possibili).

È possibile anche passare da un sistema di numerazione all'altro.

Ricordando che l'operazione che ci permette di fare i raggruppamenti è la divisione, per trasformare in base 2 il numero 14 espresso in base 10, dobbiamo dividere 14 per 2:

- $14 : 2 = 7$  Resto 0 (zero) ottengo 7 raggruppamenti da 2 unità e ho zero unità di resto.

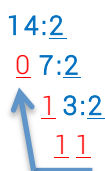
Dividiamo ancora 7 per raggruppamenti di 2:

$$7 : 2 = 3 \text{ Resto } 1$$

Infine:

$$3 : 2 = 1 \text{ Resto } 1$$

Riepiloghiamo le divisioni per 2:


$$\begin{array}{r} 14:2 \\ 0 \ 7:2 \\ \underline{1} \ 3:2 \\ \underline{1} \ 1 \end{array}$$

Per scrivere il numero in base 2 ora scrivo l'ultimo quoziente e via via, seguendo la freccia, tutti i resti:

- $14_{10} = 1110_2$

Quindi 14 in base 10 corrisponde al numero 1110 (si pronuncia uno uno uno zero) in base 2.

Per fare una verifica possiamo scrivere:

- $0 \times 1 + 1 \times 2 + 1 \times 4 + 1 \times 8$

Ossia nel sistema binario la prima posizione partendo da destra vale 1, la seconda 2, la terza 4 e la quarta 8, ecc.

Facciamo un altro esempio, vediamo a quale numero in base dieci corrisponde il numero 1010 in base 2:

- $(1010)_2 = 0 \times 1 + 1 \times 2 + 0 \times 4 + 1 \times 8 = (10)_{10}$

### 1.1.3 Dal bit al byte

Ma come fa un computer a gestire tante informazioni con due soli simboli? In realtà, un bit rappresenta solo il "mattoncino primario" di un dispositivo digitale. I bit non si trovano da soli. Essi sono uniti in gruppi di otto elementi, denominati byte.

Ciascun bit può assumere lo "stato" di spento o acceso, tradotti nei valori "0" e "1". Complessivamente, un byte, ovvero una serie di 8 bit, può assumere sino a 256 combinazioni. Moltiplicando infatti il numero 2 per se stesso 8 volte avremo:

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$4 \times 2 = 8$$

$8 \times 2 = 16$

$16 \times 2 = 32$

$32 \times 2 = 64$

$64 \times 2 = 128$

$128 \times 2 = 256$

Duecentocinquantasei saranno, quindi, le possibili informazioni che un singolo byte è in grado di modificare.

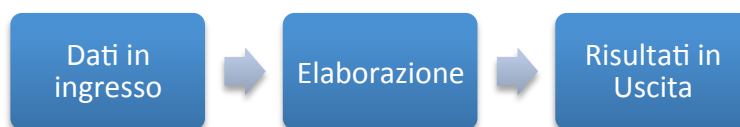
Ovviamente, il byte ha i suoi multipli:

- 1 Kilobyte = 1024 byte (circa 1000 Byte)
- 1 Megabyte = 1.048.576 byte (circa un milione di Byte)

## 1.2 Approccio agli algoritmi e ai diagrammi di flusso

### 1.2.1 Concetto di algoritmo

Per capire come funziona un computer, incominciamo a considerarlo come un "blocco" che riceve dall'esterno delle informazioni (INPUT), effettua un'elaborazione e restituisce dei risultati (output).



Concentriamoci sull'elaborazione per farci un'idea del modo in cui un computer interpreta i problemi. Pertanto è necessario introdurre il concetto di algoritmo, che può essere rappresentato graficamente tramite un diagramma di flusso.

Il termine algoritmo, che deriva dal nome di un matematico persiano, si riferisce alla serie di operazioni elementari da compiere per risolvere un determinato problema.

Le istruzioni di un algoritmo, però, devono essere:

1. ben precise;
2. elementari
3. in numero finito

### 1.2.2 Analisi di un semplice algoritmo

Immaginiamo una normale azione quotidiana, come ad esempio effettuare una telefonata.

La nostra mente in genere guida un'azione, come appunto fare una telefonata, senza riflettere sulle singole azioni che la costituiscono per raggiungere il risultato finale.

Elenchiamo le fasi di questa azione:

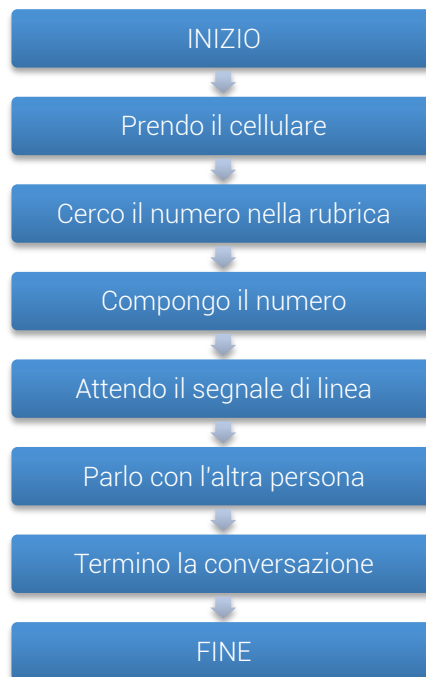
1. prendo il cellulare

2. chiamo la persona che mi interessa
3. parlo

Proviamo a scomporre alcune fasi in altre fasi ancora più elementari:

1. prendo il cellulare
2. cerco il numero all'interno della rubrica del telefono
3. compongo il numero
4. attendo il segnale di linea
5. parlo con l'altra persona
6. termino la conversazione

Una volta definita questa prima sequenza, siamo già in grado di realizzare un primo diagramma che illustrerà il nostro algoritmo:



### 1.2.3 Concetto di diagramma di flusso (flow chart)

I diagrammi di flusso rappresentano graficamente il flusso di un algoritmo e, indipendentemente dall'operazione da rappresentare, devono sempre avere:




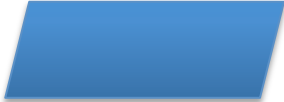

- > un blocco iniziale
- > il blocco di elaborazione
- > un blocco finale

La lettura di un diagramma procede in modo sequenziale:

1. si parte dal blocco iniziale
2. si segue la freccia in uscita
3. si raggiunge il blocco successivo e si esegue l'operazione indicata
4. si prosegue in questo modo fino a quando si raggiunge il blocco finale.

I diagrammi di flusso, denominati anche in inglese "flow chart", sono importanti perché ogni elemento, corrispondente ad una determinata azione da eseguire, viene rappresentato da una specifica forma. Ad

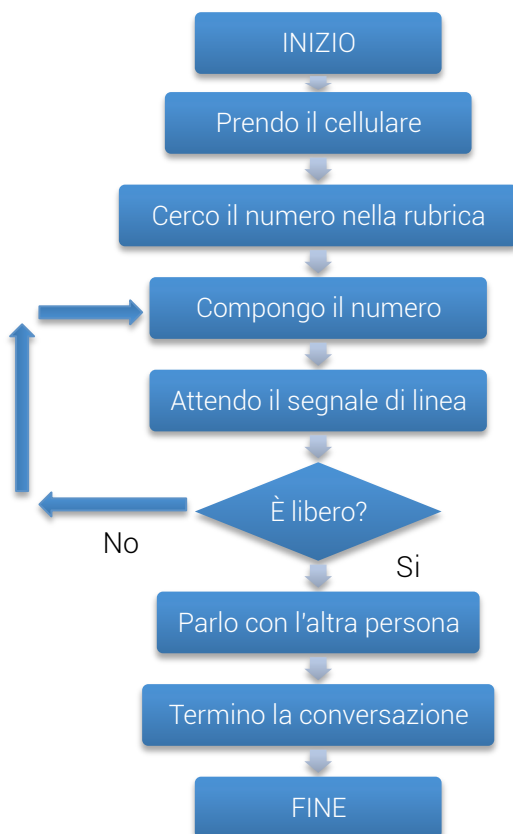
esempio, negli algoritmi precedentemente descritti, si è fatto uso di una forma a rombo per indicare una fase decisionale del processo. Le forme impiegate nella realizzazione di un flow chart sono diverse; di seguito elenchiamo le più importanti:

AZIONE	FORMA CORRISPONDENTE
Elaborazione dati	
Elaborazione alternativa	
Decisione (If...then)	
Inserimento dati	
Inizio e fine processo (Start-Stop)	

Nell'esempio della telefonata, il diagramma di flusso era sequenziale, ossia ad ogni azione seguiva necessariamente una sola azione, quindi non bisognava mai fare una scelta.

L'elemento a forma di rombo, invece, dà la possibilità di seguire un percorso o un altro.

Riprendiamo l'esempio di prima ed inseriamo un elemento a forma di rombo:

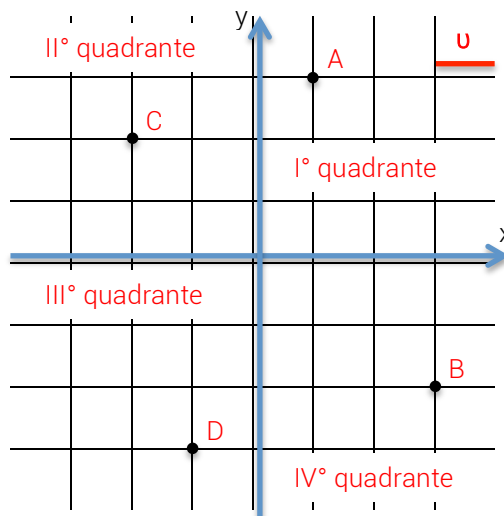


### 1.3 Rappresentazione e risoluzione di semplici problemi

#### 1.3.1 Concetto di diagramma

Il diagramma è generalmente costituito da una rappresentazione grafica di dati e descrive una situazione in modo chiaro e comprensibile.

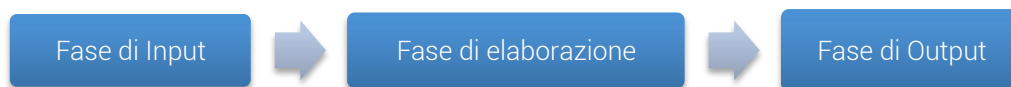
Un tipo di diagramma molto diffuso è quello cartesiano, costituito da due rette che si incrociano in un punto e formano quattro angoli retti:



In ogni quadrante è possibile individuare la posizione di un punto grazie alle sue coordinate, ossia ai due valori che il punto assume sull'asse orizzontale x e sull'asse verticale y (A = 1,3).

### 1.3.2 I diagrammi e l'informatica

In informatica, i diagrammi possono essere utili per capire come risolvere problemi di vario tipo; lo stesso schema di funzionamento di un computer, come abbiamo già visto, può essere rappresentato per mezzo di un diagramma:



Nell'immagine, il diagramma rappresentato illustra in sequenza le fasi di funzionamento del computer:

1. Inserimento dati da elaborare (fase di input);
2. il computer elabora i dati immessi (fase di elaborazione);
3. il computer restituisce i dati elaborati sotto forma di informazioni (fase di output).

È questo un diagramma di tipo "descrittivo", cioè serve a descrivere un processo. Esso può essere utilizzato per rappresentare le fasi di funzionamento di una "macchina". Esiste anche un altro tipo di diagramma che permette di capire e risolvere un problema che richiede dei calcoli: queste rappresentazioni sono conosciute come "macchine di calcolo". Per comprenderne il funzionamento e l'utilità, analizziamo il seguente esempio.

"Mario ha dodici biglie di vetro; Luca ne ha nove. Quante sono complessivamente le biglie possedute dai due compagni?"

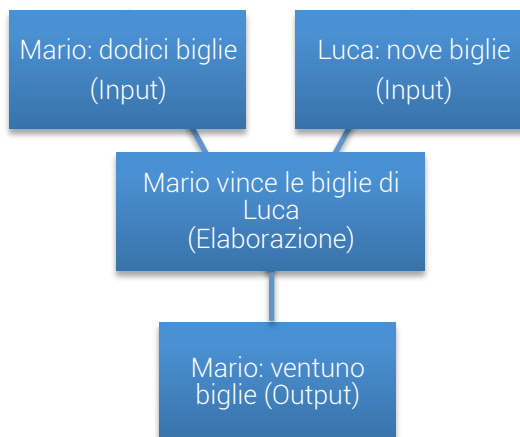
Proviamo innanzitutto ad analizzare la situazione descritta secondo metodi informatici.

**dato 1:** dodici sono le biglie che ha Mario

**dato 2:** nove sono le biglie che ha Luca

**dato 3:** (da calcolare): le biglie possedute complessivamente dai due compagni

Possiamo utilizzare il seguente diagramma per rappresentare questa situazione:



In ambito informatico possiamo scrivere:

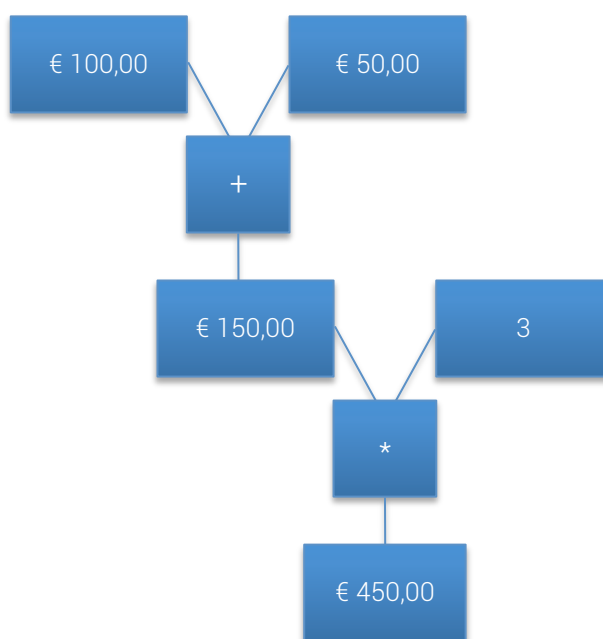
- > 12, 9: dati di input
- > Mario vince le biglie di Luca: elaborazione
- > 21: dato di output derivante dalla elaborazione dei dati di input

### 1.3.3 Diagramma di risoluzione di un semplice problema

Analizziamo la seguente situazione:

Mario possiede 100 Euro, che aggiunge a 50 Euro precedentemente messi da parte. Per il suo compleanno, il padre si impegna a triplicare la cifra che ha Mario. Alla fine, Mario avrà \_\_\_\_ Euro.

Disegniamo il diagramma, ossia rappresentiamo il metodo grafico per risolvere il problema.



Pertanto, con i dati assegnati e con le operazioni necessarie, siamo riusciti a risolvere graficamente il problema assegnato.

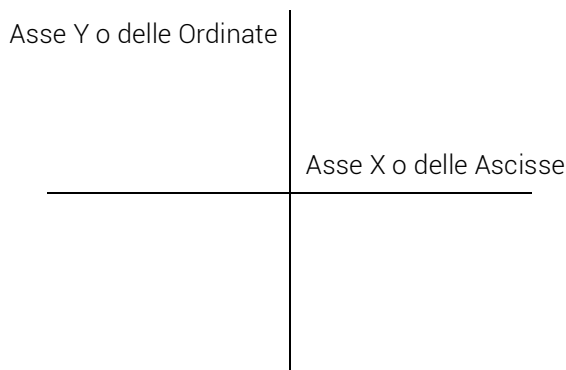
## 1.4 Classificazione delle informazioni

### 1.4.1 Definizione di grafico e di tabella

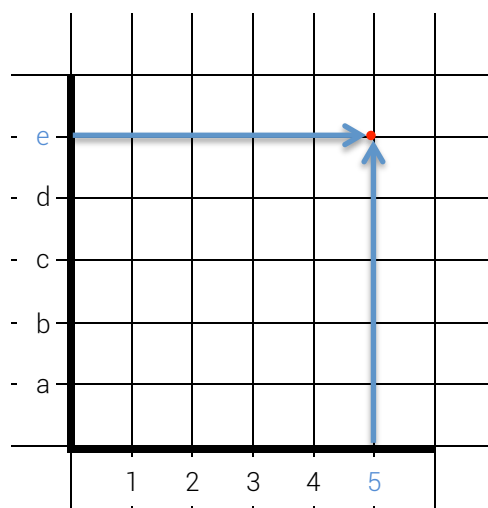
I grafici e le tabelle sono una forma di rappresentazione dei dati molto utilizzata perché facilitano la comprensione.

Per essere chiari, questi devono essere realizzati in modo da semplificare la lettura con commenti e indicazioni.

Riprendiamo l'argomento degli assi cartesiani, che sono due segmenti che si incontrano in un punto centrale e formano quattro angoli retti.



Il segmento orizzontale (asse X) viene definito "asse delle ascisse" o, in ambito geografico, "latitudine" (paralleli), e il segmento verticale (asse Y) viene definito "asse delle ordinate" (meridiani). A questo punto possiamo definire la posizione di un punto, vedendo dove si incontrano i due segmenti che partono dai due assi (X, Y):



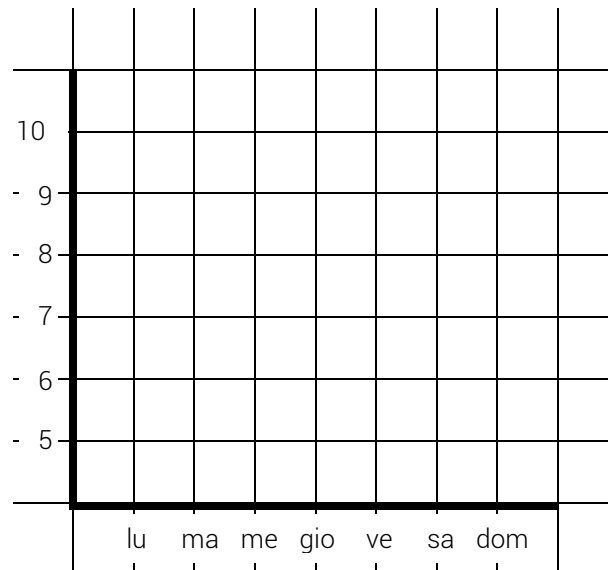
Si tratta di un vero e proprio "indirizzo" del punto rosso, che è definito dalla combinazione della lettera e del numero riportati sull'asse orizzontale e sull'asse verticale, vale a dire "5e". È un po' come l'antico gioco della battaglia navale, in cui ciascun giocatore indica al proprio avversario le coordinate di tiro sperando che in quel punto della mappa di gioco ci sia una nave avversaria. Una tabella, invece, è formata da un insieme di celle individuate all'intersezione di una riga e di una colonna.

### 1.4.2 Rappresentare una serie di dati mediante grafici e tabelle

Proviamo a considerare una tabella del sonno, in cui si deve scrivere l'orario in cui si va a letto e l'orario in cui ci si sveglia per ogni giorno della settimana:

Giorno	Vado a dormire alle ore	Mi sveglio alle ore	Totale ore sonno
Lunedì	23,00	07,00	8
Martedì	23,30	07,00	7,50
Mercoledì	00,00	07,00	7
Giovedì	22,30	07,00	8,50
Venerdì	00,00	07,30	7,50
Sabato	01,30	09,00	8,50
Domenica	00,00	10,00	10

A questo punto, possiamo disegnare il diagramma relativo a questi dati:



Nel diagramma, sull'asse orizzontale (asse x o delle ascisse) sono riportati i giorni della settimana, mentre sull'asse verticale (asse y o delle ordinate) sono riportate le ore di sonno registrate (massimo 10).

Se disegniamo delle linee verticali che partono dall'asse orizzontale e che hanno lunghezza uguale al numero di ore di sonno e poi uniamo le punte delle linee, abbiamo realizzato il grafico relativo alla tabella iniziale:

