

il Pensiero Computazionale

come pensa un informatico quando affronta un problema?

capacità di **formulare problemi**
in modo che

ammettano **soluzioni** rappresentabili
in un **formato eseguibile**

da un **uomo o da una macchina** o una
combinazione dei due
(an *information processing agent*)

Non Neutrale!

orienta la formulazione di un problema non verso la sua analisi o
comprensione, ma **verso una soluzione operativa, effettiva**



cioè scritta in un formato eseguibile in
modo automatizzabile

S. Crafa A.A. 23/24

il Pensiero Computazionale

come pensa un informatico quando affronta un problema?

capacità di **formulare problemi**
in modo che

ammettano **soluzioni**
in un **formato eseguibile**

La **programmazione** è un modo di
esercitare ed allenare questa **capacità**
di formulare e risolvere i problemi
(come la scrittura è un modo per
esprimere i propri pensieri)

una **serie di strumenti mentali** che ci consentono di

passare da un'idea alla sua realizzazione

- pensare in **maniera creativa** alle possibili soluzioni
- approccio **trial-and-error** (*esplorare e sperimentare*)
- iniziare **programmando** un prototipo

S. Crafa A.A. 23/24

soluzioni in formato eseguibile

- Un **programma** consiste di **una serie di istruzioni** che spiegano alla macchina come effettuare una computazione
 - es. un calcolo matematico, una elaborazione di un testo, un'operazione grafica, la riproduzione di un filmato...
- il **linguaggio di programmazione** è progettato per essere conciso e **non ambiguo**:
 - poche istruzioni con significato univoco e **indipendente dal contesto**
 - esclusa molta dell'espressività del linguaggio naturale, considerando *preciso solo ciò che è formalizzabile* in uno specifico linguaggio di programmazione.

S. Crafa A.A. 23/24

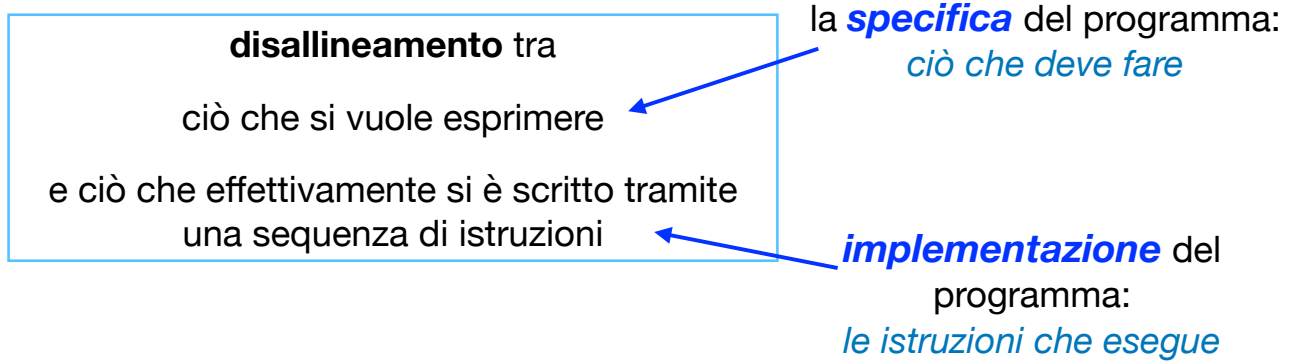
soluzioni in formato eseguibile

- Un **programma** consiste di **una serie di istruzioni** che spiegano come effettuare una computazione
- Un programma può contenere 2 tipi di **errori**:
 - **errori di sintassi**: istruzioni sintatticamente errate o struttura delle "frasi" non corretta (analogo ad errore di ortografia).
 - sono **individuati dalla macchina**, che non sa come procedere
 - **errori di semantica (o di logica)**: istruzioni sintatticamente corrette, quindi **eseguite dalla macchina**, ma **portano ad un risultato scorretto**.
 - il programmatore ha ragionato in modo errato, **sbagliando l'idea risolutiva** (e.g. errato modo di calcolare la media esami)
 - il ragionamento del programmatore è corretto, ma **nella fase di "codifica del ragionamento"**, ha inserito nel software delle istruzioni che non rappresentano correttamente il suo ragionamento.

S. Crafa A.A. 23/24

errori di semantica (o di logica)

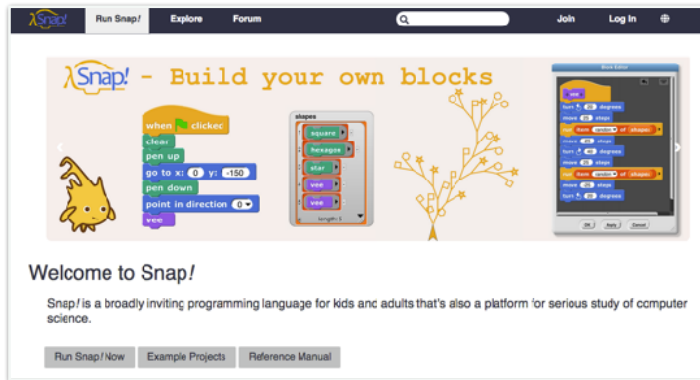
- istruzioni sintatticamente corrette, quindi **eseguite dalla macchina**, ma che **portano ad un risultato scorretto**.



S. Crafa A.A. 23/24

consideriamo un vero programma

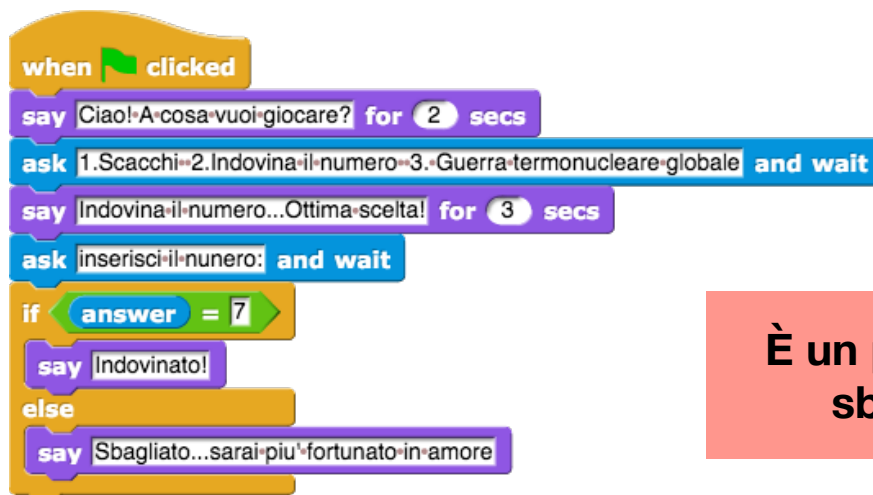
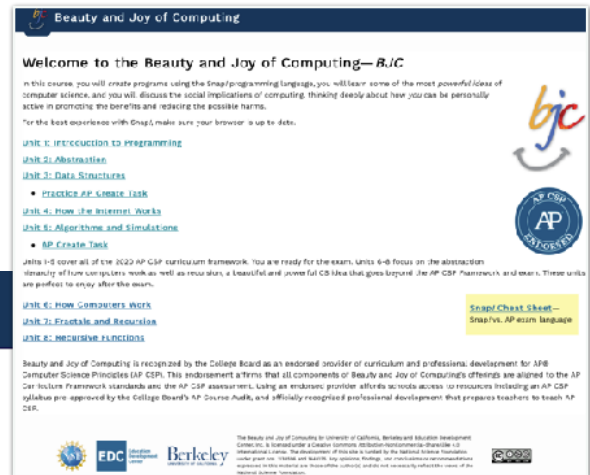
...in che linguaggio?



<https://snap.berkeley.edu>



<https://bjc.edc.org/bjc-r/course/bjc4nyc.html>

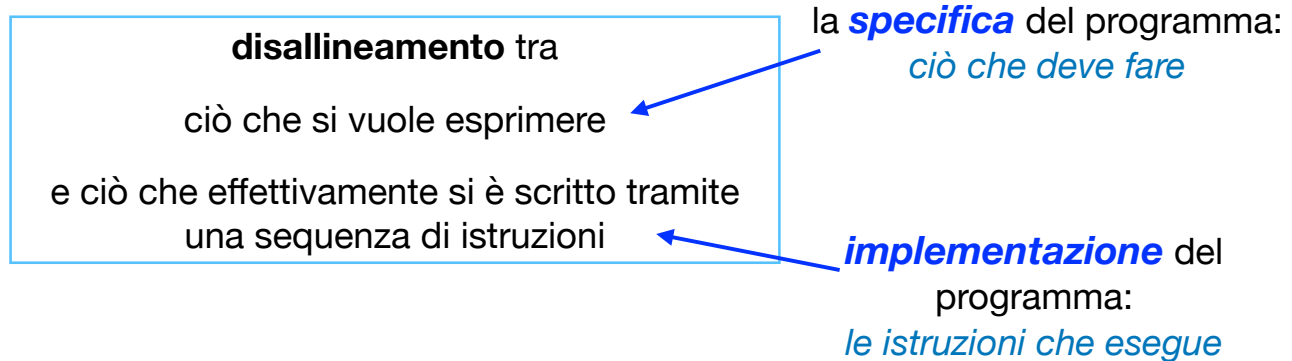


È un programma sbagliato?

```
print('Ciao! a cosa vuoi giocare?')
input('1.Scacchi 2.Indovina il numero 3.Guerra termonucleare globale')
print('Hai scelto: Indovina il numero Ottima scelta!')
answer = int(input('inserisci il numero: '))
if answer == 7 :
    print('Indovinato!')
else:
    print('Sbagliato...sarai piu fortunato in amore')
```

errori di semantica (o di logica)

- istruzioni sintatticamente corrette, quindi **eseguite dalla macchina**, ma che **portano ad un risultato scorretto**.



Come si individuano gli errori semantici ?

- spesso non è facile accorgersi che il risultato finale non è corretto!
- possono rivelarsi solo nel caso di particolari input,
es. software di calcolo tasse può dare risultato errato solo quando inserisco mese 04 senza 0.



S. Crafa A.A. 23/24

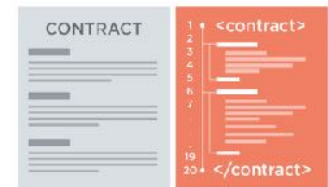
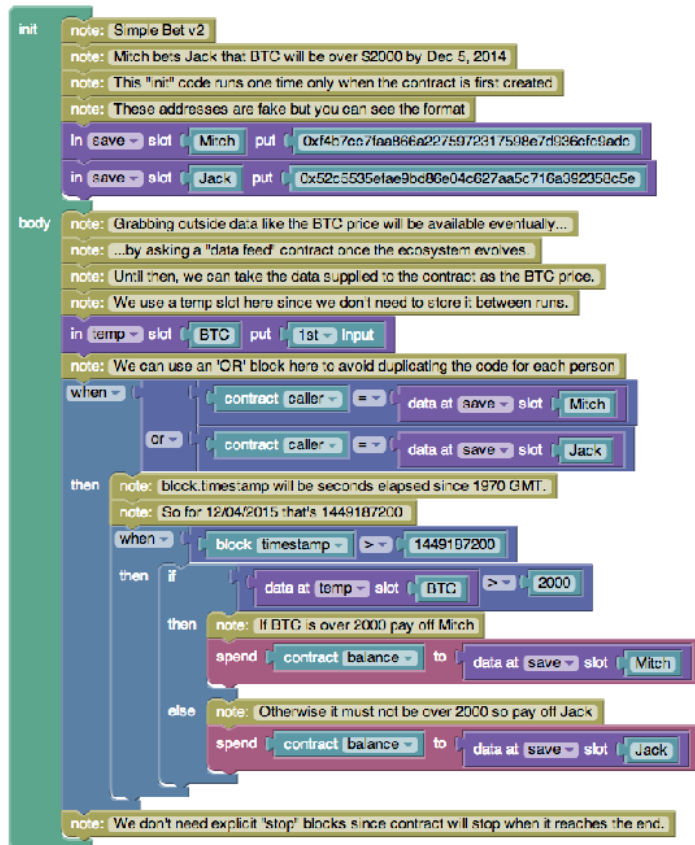
l'hacking di The DAO



- Giugno 2016, **The DAO** è un fondo di investimento completamente automatizzato su Ethereum di 150 m\$ e 11.000 persone: un partecipante sfrutta una vulnerabilità del codice per trasferire 50m \$ al fondo Dark DAO.
- **Secondo il contratto legale della DAO tutte le regole sono definite dal codice stesso**

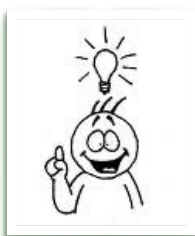
- quindi c'è **incertezza legale** se questo era un **furto** oppure una **feature** del codice.

S. Crafa A.A. 23/24



future su bitcoin (BTC)

linguaggio Marlowe
per scrivere smart contracts finanziari
sulla piattaforma Cardano



“Specifica del problema”

“Algoritmo risolutivo”

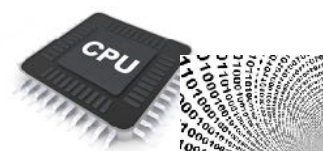
Software

editor

compilatore

Sistema Operativo

Hardware



Algoritmo vs Software

- rappresenta la **logica di funzionamento** di un programma
- è scritto in pseudo-codice, anche a parole o con diagrammi

(es. *algoritmo che assegnava i colori-Covid alle regioni in autunno 2020*)

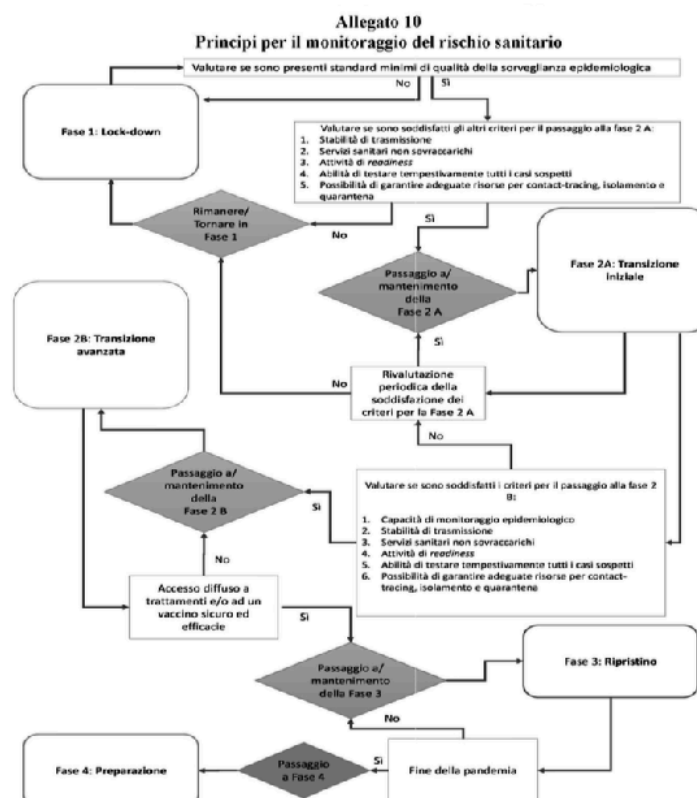
- è il **codice** di un programma, cioè una sequenza di istruzioni scritte in un preciso linguaggio di programmazione
- è **eseguibile da un computer**.

S. Crafa A.A. 23/24

27-4-2020

GAZZETTA UFFICIALE DELLA REPUBBLICA ITALIANA

Serie generale - n. 108



Algoritmo vs Software

- Un **algoritmo** è "*una sequenza non ambigua di passi che determina la procedura di soluzione di un problema*".
 - Ma questa sequenza di passi può essere descritta in qualsiasi modo (es. una ricetta per la pasta alla carbonara scritta a parole)
- L'algoritmo va **implementato**, cioè tradotto in un **software**: una sequenza di **istruzioni** scritte in un **linguaggio di programmazione**:
 - solo questa traduzione è *pienamente non ambigua ed effettivamente eseguibile*.
 - lo stesso algoritmo può essere implementato in modi diversi e in linguaggi diversi

S. Crafa A.A. 23/24

Algoritmo

- Un algoritmo è scritto con **qualche linguaggio adatto** ad esprimere "*sequenze di passi che portano alla soluzione*".
 - Ci sono **tanti linguaggi possibili**, come i diagrammi di flusso, o i sofisticati diagrammi UML, o varie forme di **pseudo-codice**
- Anche il linguaggio naturale si presta a scrivere un algoritmo. Usando però
 - un elenco di frasi brevi e chiare, e
 - i costrutti "se ... allora ... altrimenti ..." e "ripeti ..."

```
istruzione A
istruzione B
se ..... allora istruzione C1
                altrimenti istruzione C2
istruzione D
```

```
istruzione A
ripeti .....
        istruzione B
istruzione C
```

```
ripeti ..2.. volte
ripeti finché ..elenco non è vuoto..
ripeti per ogni giorno di settimana
```

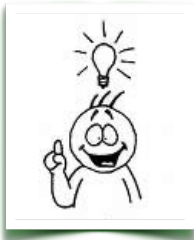
S. Crafa A.A. 23/24

Esempio

Specifica:

- un programma che, dato un elenco di prezzi di articoli, dice il totale della spesa

Algoritmo:



IDEA :

*considero un prezzo per volta e tengo traccia del **totale parziale**, che è la somma dei prezzi già considerati.*

S. Crafa A.A. 23/24

Esempio

Specifica:

- un programma che, dato un elenco di prezzi di articoli, dice il totale della spesa

Algoritmo:

INPUT : elenco di prezzi in euro **OUTPUT** : totale della spesa in euro

- tieni traccia del totale, usando una variabile di nome *tot* che all'inizio ha valore 0
- **ripeti** **finché** l'elenco non è esaurito:
 - prendi il valore attuale di *tot*
 - sommaci il prossimo numero nell'elenco
 - aggiorna il valore di *tot* con la somma appena calcolata
- restituisci il valore della variabile *tot*

S. Crafa A.A. 23/24

Esempio

Specifica:

- un programma che, dato un elenco di prezzi di articoli, dice il totale della spesa

Algoritmo:

Algoritmo tipico per sommare una serie di numeri

INPUT : elenco di prezzi in euro **OUTPUT** : totale della spesa in euro

- tieni traccia del totale, usando una variabile di nome *tot* che all'inizio ha valore 0
- **ripeti** **finché** l'elenco non è esaurito:
 - prendi il valore attuale di *tot*
 - sommaci il prossimo numero nell'elenco
 - aggiorna il valore di *tot* con la somma appena calcolata
- restituisci il valore della variabile *tot*

esempio di **esecuzione** di algoritmo:

INPUT: [10, 3.50, 11, 17.99, 22]

tot = 0

tot = 10 (0+10)

tot = 13.50 (10+3.50)

tot = 24.50 (13.50 + 11)

tot = 42.49 (24.50 + 17.99)

tot = 64.49 (42.49+22)

OUTPUT : 64.49

Esempio

Specifica:

- un programma che, dato un elenco di prezzi di articoli, **stampa lo scontrino** della spesa

Algoritmo:

INPUT : elenco di prezzi in euro **OUTPUT** : **scontrino** della spesa in euro

- tieni traccia del totale, usando una variabile di nome *tot* che all'inizio ha valore 0
- **ripeti** **finché** l'elenco non è esaurito:
 - prendi il prossimo numero dell'elenco e memorizzalo in una variabile di nome *x*
 - stampa sullo scontrino il valore di *x* e vai a capo
 - prendi il valore attuale di *tot*
 - sommaci il valore di *x*
 - aggiorna il valore di *tot* con la somma appena calcolata
- stampa sullo scontrino la scritta "totale spesa="
- stampa sullo scontrino il valore della variabile *tot*

Esemp

Specifica:

- un programma che, dato un elenco di prezzi, stampi uno **scontrino** della spesa

Algoritmo:

INPUT : elenco di prezzi in euro **OUTPUT** : scontrino

- tieni traccia del totale, usando una variabile di nome *tot* che all'inizio ha valore 0
- **ripeti** **finché** l'elenco non è esaurito:
 - prendi il prossimo numero dell'elenco e memorizzalo in una variabile di nome *x*
 - stampa sullo scontrino il valore di *x* e vai a capo
 - prendi il valore attuale di *tot*
 - sommaci il valore di *x*
 - aggiorna il valore di *tot* con la somma appena calcolata
- stampa sullo scontrino la scritta "totale spesa="
- stampa sullo scontrino il valore della variabile *tot*

esempio di **esecuzione** di algoritmo:

INPUT: [10, 3.50, 11, 17.99, 22]

```
tot = 0
tot = 10 (0+10)
tot = 13.50 (10+3.50)
tot = 24.50 (13.50 + 11)
tot = 42.49 (24.50 + 17.99)
tot = 64.49 (42.49+22)
```

OUTPUT :

```
10
3.50
11
17.99
22
totale spesa = 64.49
```

S. Crafa A.A. 23/24

Esercizio **IMPORTANTE** su Algoritmi

Scrivere i seguenti algoritmi, trovando un linguaggio efficace.

Inserire le soluzioni, eventualmente incomplete, sul forum Moodle e discuterne

ATTENZIONE: questo è un esercizio difficile perché non è facile trovare il modo più adatto di scrivere questi algoritmi. Lo scopo vero dell'esercizio, non è arrivare alla soluzione ma **esplorare quali difficoltà si incontrano provandoci**.

- Scrivere l'algoritmo per preparare la carbonara. Elencare all'inizio gli ingredienti (cioè i dati di input) e poi descrivere l'algoritmo per realizzare il piatto.
- Scrivere l'algoritmo per calcolare la media dei voti degli esami sostenuti.
- Il gioco "pari o dispari" ammette 2 giocatori: il primo giocatore scommette sempre su "pari", mentre il secondo scommette sempre su "dispari". Ogni giocatore sceglie un numero, e se la somma dei due numeri è pari vince il primo giocatore, altrimenti vince il secondo. Scrivere l'algoritmo di un programma che gioca a "pari o dispari"
- Scrivere algoritmo di un torneo sportivo ad eliminazione diretta a partire dai quarti di finale. Cioè ci sono 4 coppie di squadre che giocano, le vincitrici vanno alle semifinali e le vincitrici alla finale.
- Scrivere un algoritmo per l'elezione del capoclasse.

S. Crafa A.A. 23/24