Programming Fundamentals

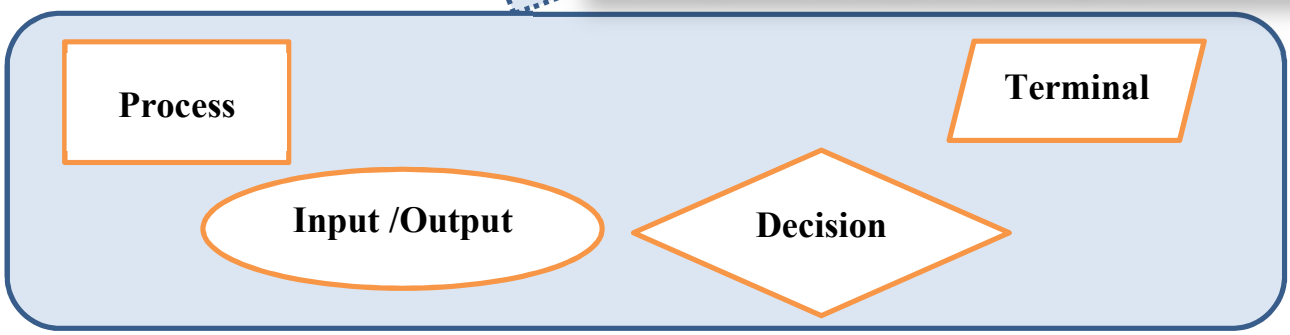
Programming is a problem-solving procedure. It is the act of designing and implementing computer programs.

A ***program***, is a list of instructions for the computer to follow to accomplish a task. The programs that computer executes are called *software*.

Programming (*Software Development*) follows a six-step process:

1. **Program specification/ definition/ analysis:** describing the problem (objectives, outputs, input, and processing requirments)
2. **Program design:** making a plan (***Flowchart***, ***Pseudocode***)
3. **Program code**: Coding "*speaking the language of the computer*
4. **Program test**: Debugging "*Getting rid of errors*"
5. **Program documentation**
6. **Program maintenance**

Pseudocode ("soo-doh-code") is an outline of the logic of the program. It is an informal description of a sequence of steps for solving a problem.



Program Code: Write the program using the appropriate computer language.

A ***programming language***, uses a collection of symbols, words & phrases that instruct a computer to perform specific operations.

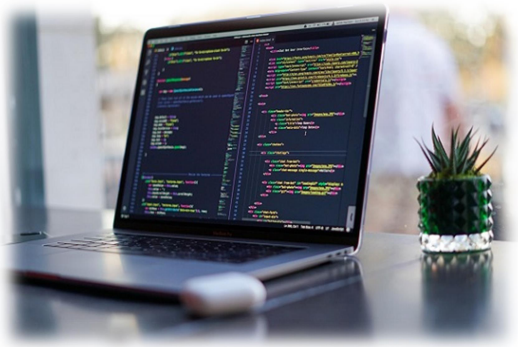
Program Test: *Debugging*, the process of testing & eliminating errors. (syntax & logic errors)

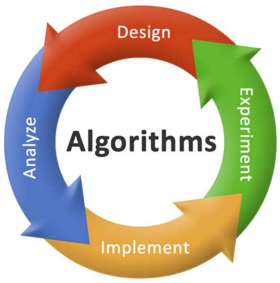
Program Documentation: Users need to know how to use the program.

Program Maintenance: To ensure that current programs are operating error-free, efficiently. & effectively.

Programming Language Generations

- Low-level languages are closer to the 0s & 1s
- High-level languages. are closer to the languages of humans





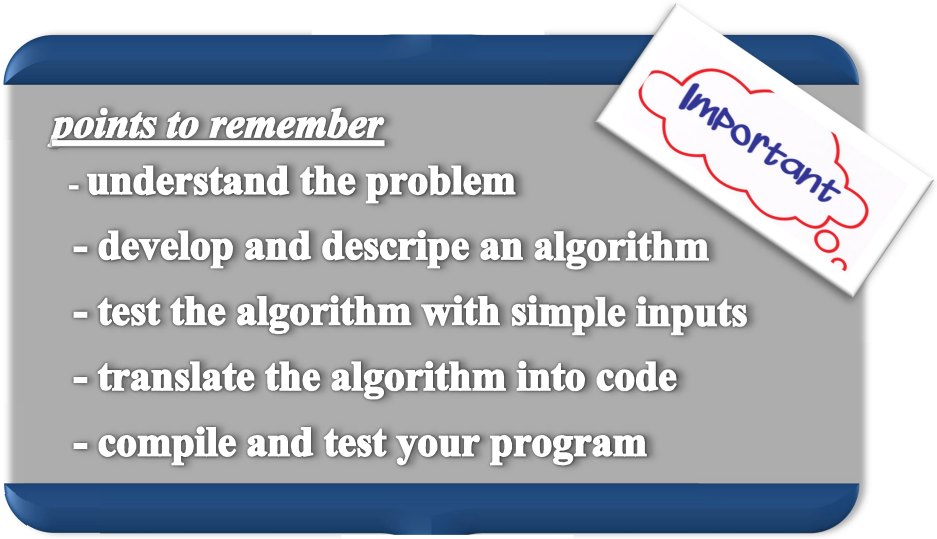
Making a Plan : Algorithm Development

Algorithm, is a set of specific sequential steps that describe in natural language exactly what the computer program must do to complete its task. (is a set of rules to solve a problem).

Different algorithms could be used to complete the same task. (one way may be better). A *program* is an algorithm that has been translated (coded) into instructions for a computer.

Development Tools/ Integrated Development Environment (IDE)

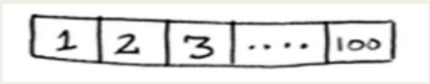
IDE is a developmental tool that helps programmers write & test their programs.



Let's Play

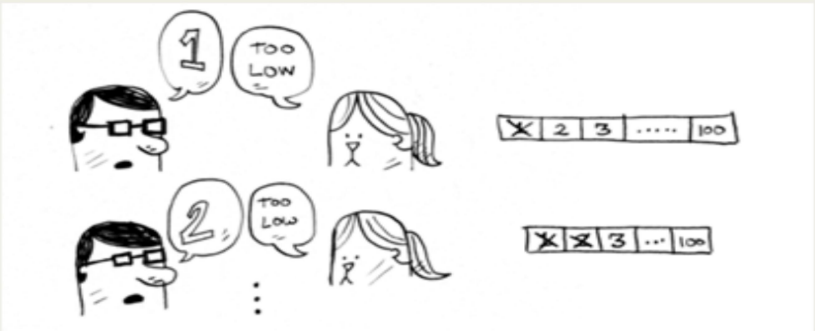
Guess a Number

- I'm thinking of a number between 1 and 100.



- You have to try to guess my number in the fewest tries possible.
- With every guess, I'll tell you if your guess is too low, too high, or correct.

- Suppose you start guessing like this: 1, 2, 3, 4 ....

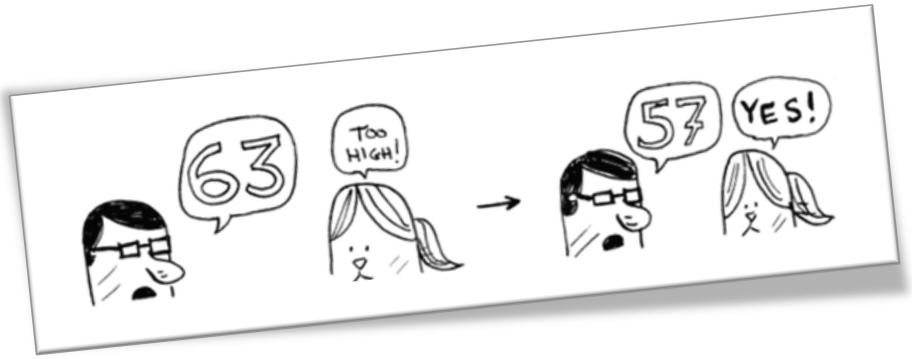


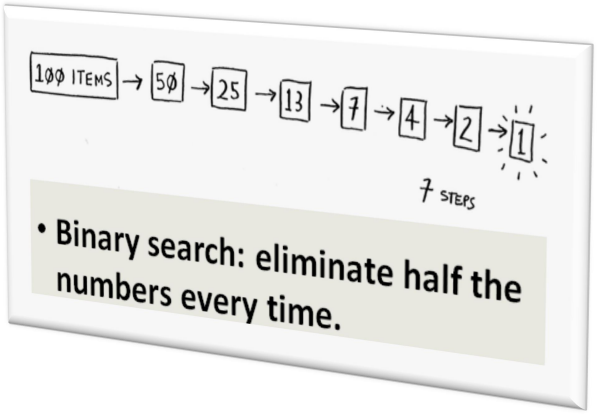
- This is *simple search.* With each guess, you're eliminating only one number. If my number was 99, it could take you 99 guesses to get there!

A better way to search

Here's a better technique. Start with 50.

Too low, but you just eliminated *half* the numbers! Now you know that 1–50 are all too low. Next guess: 75.



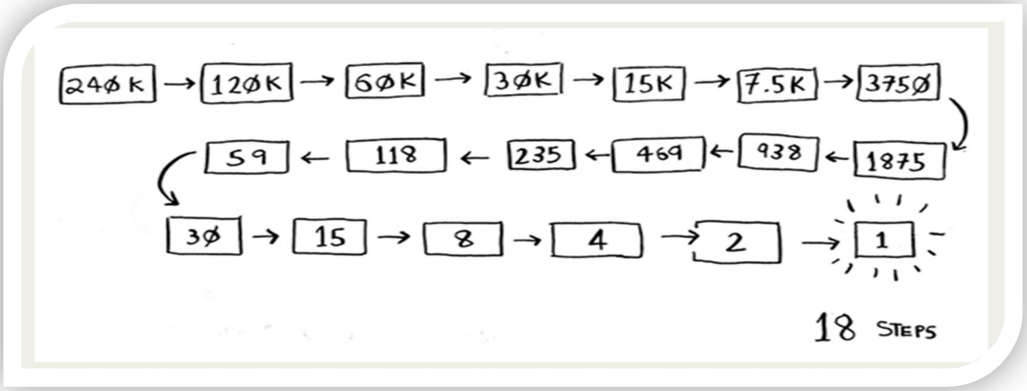


Looking for a word in the dictionary

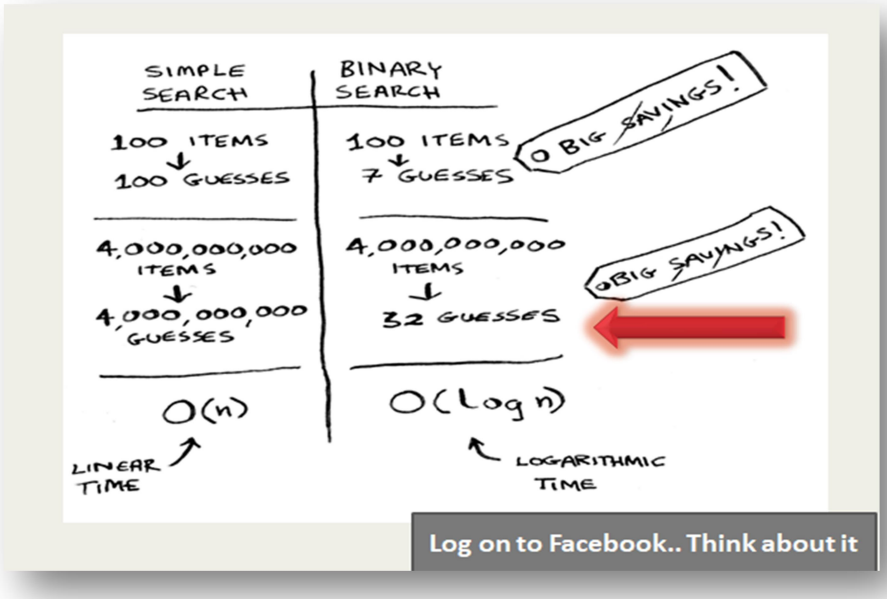
- Suppose you're looking for a word in the dictionary. The dictionary has 240,000 words. *In the **worst case***, how many steps do you think each search will take?
- *Using **binary search***, how many steps do you think each search will take?

SIMPLE SEARCH: _____ STEPS

BINARY SEARCH: _____ STEPS



For binary search, you have to check log *n* elements in the worst case. For a list of 8 elements, log 8 == 3, because 2^3 == 8. For a list of 1,024 elements, log 1,024 = 10, because 2^10 == 1,024.



Examples

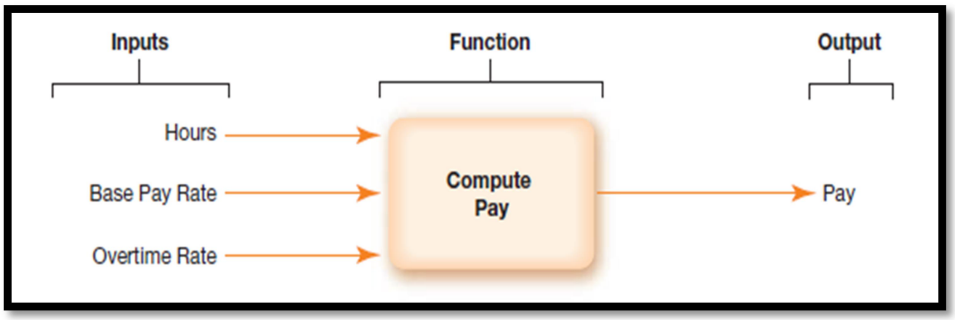
- **Add two numbers**

  Get first number
  Get second number
  Add both numbers
  Show the result

- **Calculating daily wages**

  Supposedly : An employee gets **3$** per hour of regular work (first eight hours) and **5$** per hour of overtime work (each hour worked in excess of 8 hours)

  Read the number of work hours
  If number_of_work_hours < 8
  *Total = number_of_work_hours * 3*
  *Else Total = (8*3) + ((number_of_work_hours-8)*5)*
  Print Total



- **Check the number (even or odd)**

1. *Begin*
2. *Read the value of the number*
3. *Divide number by 2 and store the reminder in var*
4. *If var is 0, go to step 7*
5. *Print "number is an odd"*
6. *Go to step 8*
7. *Print "number is an even number"*
8. *End*