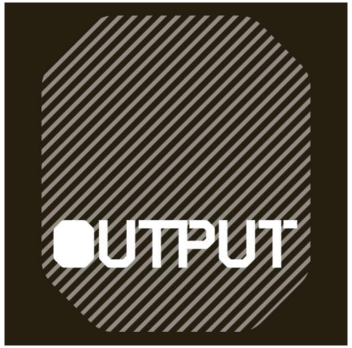
Pseudocode

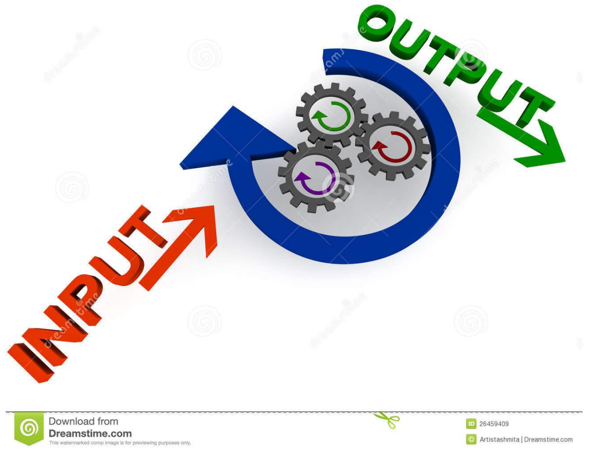
Pseudocode

- Describes algorithms in a more robust, mathematical sense then writing them step-by-step

Pseudocode

- Needs defined: Inputs, Outputs, and Assumptions
    - Help clarify what is needed and what the goal is





Decimal to Hex: Pseudocode

- Converting base 10 to hex- **Divide by 16 Algorithm**

1) Assume the number is > 0
2) Divide the number by 16, write the remainder in a stack (bottom up)- convert to hex notation if necessary
3) When the number is reduced to zero, flip the stack. This is your hex number.

Decimal to Hex: Pseudocode

- Assumptions: Number is > 0
- Input: A base 10 number
- Output: A hexadecimal number

- *These three parts MUST be included in any pseudocode you write*

Pseudocode Details

- Prompts must appear as they would appear to a user
- State destination of output (ex: Display, File)
- Each number should be a variable
  - Should represent a GENERAL number, not a specific value

Pseudocode Details

- Surround variable names with ‘<’ and ‘>’
- Make up instructions as necessary
    - “Turn on the computer”, “Vibrate phone for 1 second”

Decimal to Hex Pseudocode

```
<dec> = input

<stack> = <empty>
```

Initialize variables - set <dec> (decimal) to the input, and set the <stack> to empty (interpreted as all 0s)

Decimal to Hex Pseudocode

```
<dec> = input

<stack> = <empty>

While <dec> does not = 0:

    <dec> = <dec> / 16

    Push remainder to top of stack
```

The “While” section repeats until the condition (<dec> does not = 0) is reached

Decimal to Hex Pseudocode

```
<dec> = input

<stack> = <empty>

While <dec> does not = 0:

    <dec> = <dec> / 16

    Push remainder to top of stack

Reverse <stack>

Print <stack> to Display
```

Display the final answer on the screen by "printing" it

More pseudocode

Draw a line of *’s on the computer screen

Algorithm: The user will enter the length of the line, the computer will draw a line of that same number of *s

More pseudocode

Draw a line of *’s on the computer screen

Algorithm: The user will enter the length of the line, the computer will draw a line of that same number of *s

Assumptions: The user will enter a number

Input: A number

Output: A line of stars

Draw a line pseudocode

———

```
<size> = input
If <size> < 0
    Display “Can’t have a
        negative length”
```

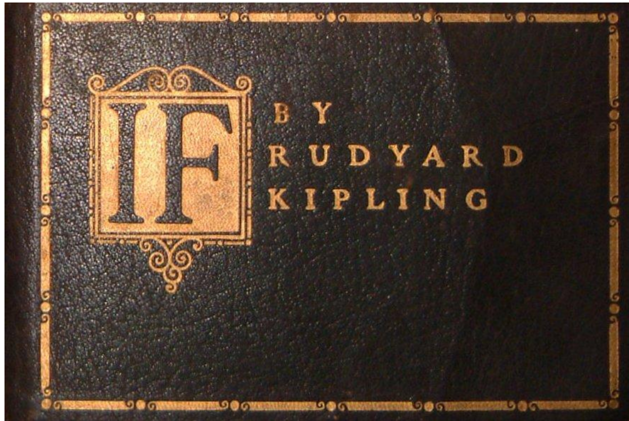
Conditional to make sure the user entered a valid number

Draw a line pseudocode

```
<size> = input
If <size> < 0
    Display "Can't have a
        negative length"
Else
    <len> = 0
    While <len> < <size>    while length is less then size...
        Print "*"           ...print a star...
        <len> = <len> + 1   ...then add one to "len"
```

Repetitions and Conditionals

- Conditionals provide the foundation for pseudocode
  - **If** something is true...complete a task
  - **Else** complete another task



Repetitions and Conditionals

- Repetitions can be added at any point in pseudocode
    - Can happen **while** a condition is waiting to become true
    - Can happen **for** a specific number of times

Pseudocode Practice

- Walk across Sinclair Lane

- Assumptions: You can follow directions and see oncoming cars.
- Input: Cross the street
- Output: Crossed the street

Pseudocode challenge

- You have to program a robot to walk from Room 205 to an unspecified room on the second floor of Curley.
- <u>Assumptions:</u> The robot is placed directly in the middle of Room 205 and knows directions (left, right, etc…). It can read classroom number signs. The user will input an existing room.
- <u>Input:</u> The room the robot will go to.
- <u>Output:</u> The robot will beep when it reaches its destination