

# Recitation 01

Legos-Unix-Git

George Mappouras

9/1/2017

# Lego

Find the algorithm:

Count from 1 to N, call each number *i*. For each *i* do:

if (**STATEMENT**)

    place some Lego brick with color C1

Otherwise

    place some other Lego brick with color C2

# Legos-Step1

Find the algorithm:

You are given 4 boards. Each board represents the algorithm for different  **$N=1,2,3,4$**

Count from 1 to  $N$ , call each number  $i$ . For each  $i$  do:

if (**STATEMENT**)

**place some Lego brick with color C1**

Otherwise

**place some other Lego brick with color C2**

# Legos-Step1

Write down the algorithm and return to your TA

**N=1,2,3,4**

Count from 1 to N, call each number *i*. For each *i* do:

if (**STATEMENT**)

    place some Lego brick with color C1

Otherwise

    place some other Lego brick with color C2

# Legos-Step2

You are given an extra board with legos.

Execute the algorithm you found for  $N=5$

Count from 1 to  $N$ , call each number  $i$ . For each  $i$  do:

if (STATEMENT)

place some Lego brick with color C1

Otherwise

place some other Lego brick with color C2

# Lego-Step3

Exchange algorithms with the team next to you.

Execute the new algorithm for  $N=5$

# Legos-Algorithms

Algorithm 1:

Count from 1 [inclusive] to  $N$  [inclusive], call each number  $i$

If  $(i + N)$  is even)

Place a  $2 \times 2$  C1 block with its lower left corner at  $(3*i, 2*i+1)$

Otherwise

Place a  $4 \times 2$  C2 block with its lower left corner at  $(3*i+1, 2*i)$  such that the long end is vertical

# Legos-Algorithms

## Algorithm 2:

Count from 1 [inclusive] to  $N$  [inclusive], call each number  $i$

    If  $(i \geq N/2)$

        Place a 4x2 C1 block with its lower left corner at  $(2*(i-1), i+2)$  w/ the long end vertical

    Otherwise

        Place a 4x2 C2 block with its lower left corner at  $(2*(i-1), 2*i)$  w/ the long end vertical



# Unix

1. **Change Directory:** `cd /path/to/my/directory`
2. What is my current directory? `pwd`
3. **List** my files in my current directory: `ls`
4. **Make new Directory:** `mkdir /path/to/my/directory/new_folder`  
`mkdir -p /path/to/my/directory/parent_folder/child_folder`

# Unix

5. **Copy** files: `cp /path/to/my/file/file1 /path/to/my/file/file_copy`

6. Cut or **Move** or rename files:

`mv /path/to/my/file/file1 /new/path/file1`

# Unix: Absolute vs Relative path

`cd /home/gm118/` (Change directory to the absolute path)

`cd ./ece551` (relative path depending on my current directory)

`cd ../` (go up one directory)

# Unix: Use with COTION

**Remove a file:** `rm ./my_file`

**Remove a **directory**:** `rmdir ./myfolder/`

`rm -rf ./myfolder/` (Delete all files and folders inside myfolder)

`-r` = Recursively, `-f` = Force

# Git Commands

1. `git pull`: Updates all your file
2. `git add file_name`
3. `git commit -m "A message to remind me what I changed!"`
4. **`git push` (Don't forget this command)**
5. **`grade exercise_title` (Don't forget this either!)**

# Git – deleted files

git status -> information about git. Needs to be clean to grade

git checkout -- deleted\_file -> Undo the deletion

git rm -> remove deleted files from repository