

## IT 112: Introduction to Programming

Dr. Manish Khare

Dr. Bakul Gohel



Lecture 2

#### **Last Lecture's Exercise**

Write an algorithm to compute the quotient and reminder when a given integer x ( $x \ge 0$ ) is divided by another integer y ( $y \ge 0$ )

Assume that only addition and subtraction are available as primitive arithmetic operators.

#### **Last Lecture's Exercise**

#### **Properties of Quotient and Reminder:**

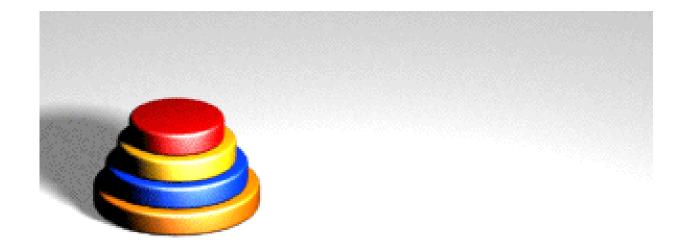
If q and r are quotient and reminder respectively obtained after dividing x by y:

- q \* y + r = x
- r < y

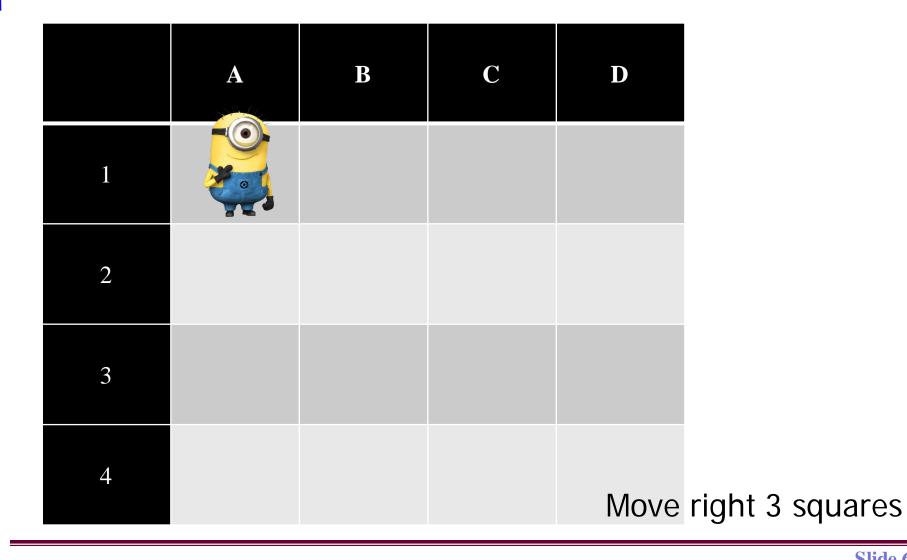
#### **Solution**

- 1. Read the values of x and y
- 2. Set value of q to 0.
- 3. While  $(x \ge y)$  do (steps 3.1 and 3.2)
  - 1.  $x \leftarrow x y$ ;
  - $q \leftarrow q + 1$
- 4. Print results: reminder is x and quotient is q.

## **Tower of Hanoi**



## How do I get Minion Stuart to move to D1?



# How do I get Minion Stuart to move to D1 then to D4?

	A	В	С	D	
1					
2					
3					
4				Move Move	right 3 squares down 3 squares

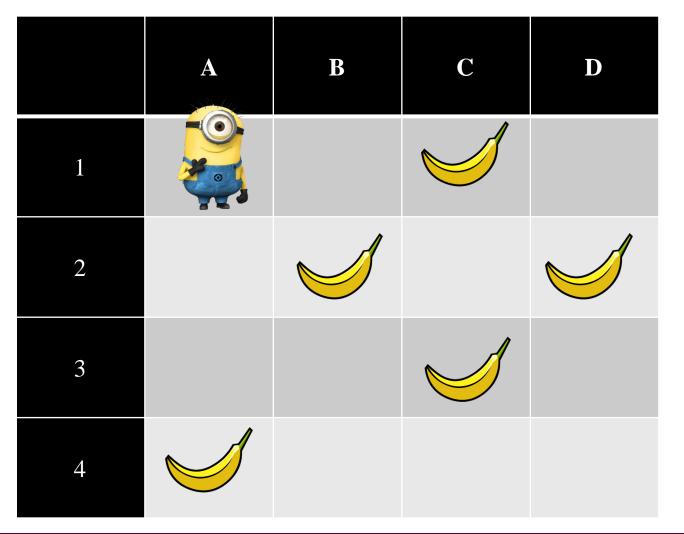
# How do I get Minion Stuart to move to D1 then to D4, then move to A4 and finally to A1?

	A	В	С	D	
1					
2					
3				Mo	ove right 3 squares ve down 3 squares Nove left 3 squares
4					Move up 3 squares

## What have we just done

- You have created a series of instructions to solve a given problem
- This is called an Algorithm
- When we write it in a list of instructions it is called **Pseudocode**
- Computer Programmers use pseudocode to help plan out the code they will need.
  - For game making
  - Creating websites
  - Control software robots / machinery
  - ANYTHING where planning is needed = pseudocode is used to layout the tasks/actions

In your workbooks I want you to think about how to get Minion Stuart to move around the squares and collect the bananas. You will need to move him and also add actions needed to pick up the items.



### **Your Turn**

Extension =
Can you add a
'question to
say if the
minion reaches
a banana?



## Algorithm Solution written in pseudocode



	A	В	C	D
1				
2				
3				
4				

- Move right 2 squares
- 2. Pick up banana
- Move right 1 square
- 4. Move down 1 square
- 5. Pick up banana
- 6. Move left 2 squares
- 7. Pick up banana
- Move down 1 square
- 9. Move right 1 square
- 10. Pick up banana
- Move down 1 square
- Move left 2 squares
- Pick up banana

## Can you ask a question?

When we think about the way Minion Stuart moves across could we ask a question as he moves from one side to the other?

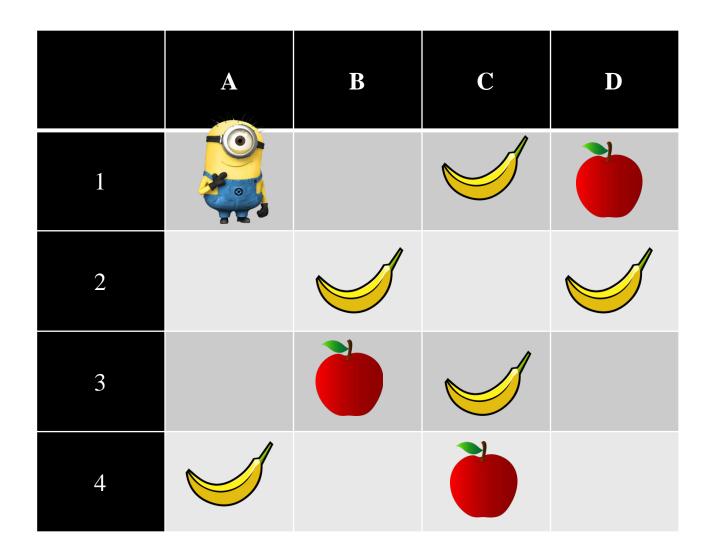
- 1. Move across 3 squares
- 2. If you reach a banana
  - a. Pick it up

Ask the question – try and start it with an 'IF'

Ask the question – try and start it with an 'IF'



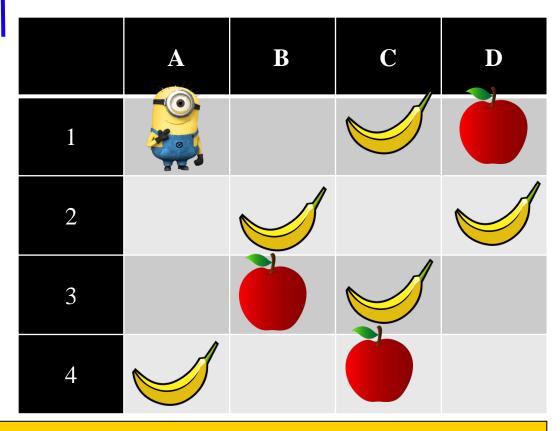
## Can we ask a question?





## Possible algorithm in pseudocode





This form of pseudocode would help a game designer plan out the code they would need to write to create it.

- 1. Move right 3 squares
- 2. IF Minion reaches a banana
  - a. THEN pick it up
- 3. IF Minion reaches an apple
  - a. THEN leave it

This would continue to cover the whole board

## Pseudocode: Keywords

- begin ----- end: begin is the first statement and end is the last statement of the pseudocode. All the instructions used in pseudocode is to be written between begin----end block.
- Read: Reading two values from the input given by the user
- Compute: The keyword compute is used for calculation of the result of the given expression.
- Print: It is used to display the output of the program
- if--- else--- endif: It is a decision construct used different action has to be taken depending on the condition.
- while---- do: This is also an iterative construct similar to do---while construct. The only difference is that in this construct, the testing statement is present at the top of set of the iterative statements.

## Pseudocode: Example

Write the pseudocode to accept two numbers and displays their sum and difference.

Begin //starting the pseudocode

Read first\_number, second\_number //input two numbers

Compute sum as first\_number + second\_number // process of adding the given two numbers

Compute difference as first\_number - second\_number //process of calculating difference of the given two numbers

Print 'The sum of the given two numbers is' sum //Output to display the sum

Print 'The difference of the given two numbers is' difference //Output to display the difference

End //End of the pseudocode

#### **Decision Structures**

- The expression A>B is a logical expression
- it describes a condition we want to test
- if A>B is true (if A is greater than B) we take the action on left
- print the value of A
- if A>B is false (if A is not greater than B) we take the action on right
- print the value of B

#### if-then-else Structure

■ The structure is as follows

```
If condition then

true alternative

else

false alternative

endif
```

Relational Operators			
Operator	Operator Description		
>	Greater than		
<	<b>Less than</b>		
=	Equal to		
≥	Greater than or equal to		
<b>≤</b>	Less than or equal to		
<b>≠</b>	Not equal to		

#### **Nested if Structure**

■ One of the alternatives within an IF—THEN—ELSE statement

□ may involve further IF–THEN–ELSE statement

## **Loop Structure**

For Loop

➤ While... Do Loop

➤ Do... While Loop

#### **Exercise**

pseudocode for compute the area of a rectangle

Begin

Get the length, l, and width, w

Compute the area = l\*w

Display the area

End

pseudocode for compute the perimeter of a rectangle:

Begin

Enter length, l

Enter width, w

Compute Perimeter = 2\*l + 2\*w

Display Perimeter of a rectangle

end

> pseudocode for Calculate Pay – sequence

Begin
input hours
input rate
pay = hours \* rate
print pay
End

**>**1

pseudocode for Calculate Pay with Overtime – selection

```
Begin
input hours, rate
if hours \leq 40 then
    pay = hours * rate
else
   pay = 40 * rate + (hours - 40) * rate * 1.5
print pay
End
```

#### Within Class Exercise

- Larger of two numbers
- Roots of a quadratic equation
- ► Grade Computation

MARKS 
$$\geq$$
 90  $\Rightarrow$  Ex  
89  $\geq$  MARKS  $\geq$  80  $\Rightarrow$  A  
79  $\geq$  MARKS  $\geq$  70  $\Rightarrow$  B  
69  $\geq$  MARKS  $\geq$  60  $\Rightarrow$  C  
59  $\geq$  MARKS  $\geq$  50  $\Rightarrow$  D  
49  $\geq$  MARKS  $\geq$  35  $\Rightarrow$  P  
34  $\geq$  MARKS