More Algorithms

Course: CPSC 1150
Instructor: Dr. Bita Shadgar

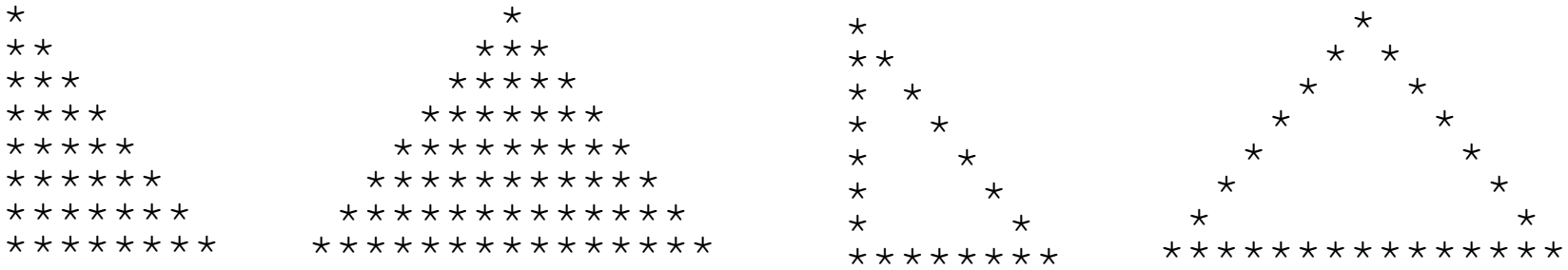
Example 2

- Write an algorithm that given a value for the height of triangle (eg. H), it draws a triangle with the given height H like following:

H

5

8



Exercise A

```
H
5    *
     **
     ***
     ****
     *****


8    *
     **
     ***
     ****
     *****
     ******
     *******
     ********
```

```
START
        get H from user input
        n ← 1
        Repeat while n ≤ H:
                print n of '*' in a new line
                n ← n + 1
END
```

Exercise B

H
5

```
    *
   ***
  *****
 *******
*********
```

8

```
       *
      ***
     *****
    *******
   *********
  ***********
 *************
***************
```

START

Input H from user input

line ← 1

Repeat while line ≤ H:

print (H-line) spaces in a new line

n ← 2*line – 1

print n of ‘*’ in the same line

line ← line + 1

END

Exercise C

H

5
```
*
**
*  *
*   *
*****
```

8
```
*
**
*  *
*   *
*     *
*       *
*         *
********
```

START

get H from user input

print * in a new line

n ← 2

Repeat while n < H:

print ‘*’ in a new line

print n-2 spaces in the same line

print ‘*’ in the same line

n ← n + 1

print n of ‘*’ in the new line

END

Exercise D

H

5

8

START

read H from user input
line ← 1
space ← H – line
print space of ‘ ‘ in a new line
print ‘*’ in the same line
line ← 2
Repeat while line < H:
    space ← H – line
    print space of ‘ ‘ in a new line
    print ‘*’ in the same line
    space ← (2*line – 1) - 2
    print space of ‘ ’ in the same line
    print ‘*’ in the same line
    line ← line + 1
print (2*line – 1) of ‘*’ in a new line

END

Example 3

Algorithm: Find average of a class for an exam.

(Define terms)

Let numOfStudents be the number of the students in the class.

Let grade be the grade of each students.

START

numOfStudents ← 0

total ← 0

Repeat for all students

read *grade* of the student.

total ← *total + grade*

Increment *numOfStudents* by 1

average ← *total / numOfStudents*

print average

END

Example 4

Algorithm: Find number of the students passed or failed in a class

(Define terms)

Let *passed* be the number of the students passed.
Let *failed* be the number of the students failed.
Let *grade* be the grade of each students in the exam.

START

passed ← 0
failed ← 0
Repeat for all students
read *grade* of the student.
If *grade* >= 50 Then
passed ← *passed*+1
Otherwise
failed ← *failed*+1
print *passed* and *failed*

END

Working on Digits of a Number

Write an algorithm that given an integer number, it finds the sum of all it digits. For example, if the number is 3056, then the algorithm prints 14.

- To separate the last digits of a number from the rest, use remainder (%) by 10
  - 3056 % 10 → 6
- To get the rest of digits use integer divide (//) by 10
  - 3056 // 10 → 305
- **Note:** finding integer division is done implicitly or explicitly in different programming languages. But use // to show integer division in the algorithms.

Similar Examples

- Write an algorithm that given a number, it prints true if the sum of its odd digits equals to its even digits. Otherwise it prints false.

For example, given 12045 it prints true, but given 3127 prints false.

Similar Examples

- Write an algorithm that gets a number and it produces another number with reverse order of the digits in the original number. The leading zeros are ignored.

Original Number	The Result
12365	56321
7601	1067
980	89
370000	73

Example 5 - Factorial

The factorial of a number **`X`**, **`X!`** is the product of all integers from 1 up to and including X

- e.g. 10! = 10(9)(8)(7)…(3)(2)(1) = ?

  4! = 4(3)(2)(1) = 24

– Write an algorithm to compute **`X!`** (use flowchart and pseudo-code to show your algorithm)

Pseudo code for Factorial

Algorithm: Find n! (Factorial of n)

(Define terms)

Let *n* be the number input by user

Let *factorial_n* be the factorial of *n*

START

1. Input *n*
2. *factorial_n* ← 1
3. Repeat while *n* > 0
   - 3.1 *factorial_n* ← *factorial_n* * *n*
   - 3.2 *n* ← *n -1*
4. print *factorial_n*

END

Test using Trace Table

n	factorial_n	Output
5	1	

Example 6 - Fibonacci

The Fibonacci sequence is the sequence of integers such that every number in the sequence is the sum of the previous two numbers in the sequence.

- e.g. 1, 1, 2, 3, 5, 8, 13, 21, …

– Write an algorithm to display the first X values of the Fibonacci sequence. (use flowchart and pseudo-code to show your algorithm)

Pseudo code for Fibonacci

Algorithm: Print first x Fibonacci numbers

(Define terms)

0.1 Let *x* be the number input defined by user.

START

1. Input *x*
2. *num1* ← 0
3. *num2* ← 1
4. *newNum* ← 1
5. *counter* ← 0
6. Print *newNum,*
7. Repeat while x>= *counter*
    - 7.1 *newNum* ← *num1* + *num2*
    - 7.2 Print *newNum,*
    - 7.3 *num1* ← *num2*
    - 7.4 *num2* ← *newNum*
    - 7.5 *counter* ← *counter* +1

END

Test using Trace Table

x	num1	num2	newNum	counter	output
5					

Is this algorithm Correct?