CSE 1061 **Introduction to Computing**

Lecture 8

Fall 2015

Science & Computing Engineering Program
The School of EE & Computing
Adama Science & Technology University

OUTLINE

Adding beepers

Triangular inequality

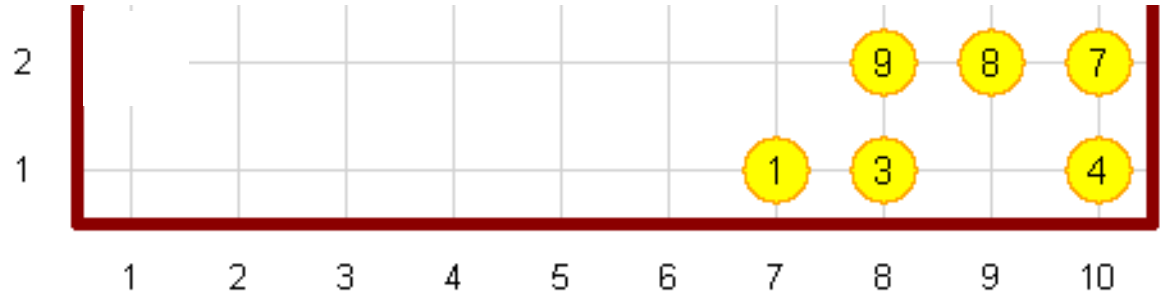
Drawing graphs

ADDING BEEPERS

ASTU

PROBLEM 19: Adding Beepers*

Given beepers in two rows of a 2D world, Hubo wants to **colle ct** the **beepers** in **each column** and **put them** at **the place in the first row**. Hubo should **move back** to the **starting p osition** and also **recover his orientation** after finishing his t ask. You may assume that beepers, if any, are initially placed in **the first two rows**, that is, the first and second rows. Not every column has beepers on two places: A column may have no places with beepers and other column may have one place with beepers as shown below:

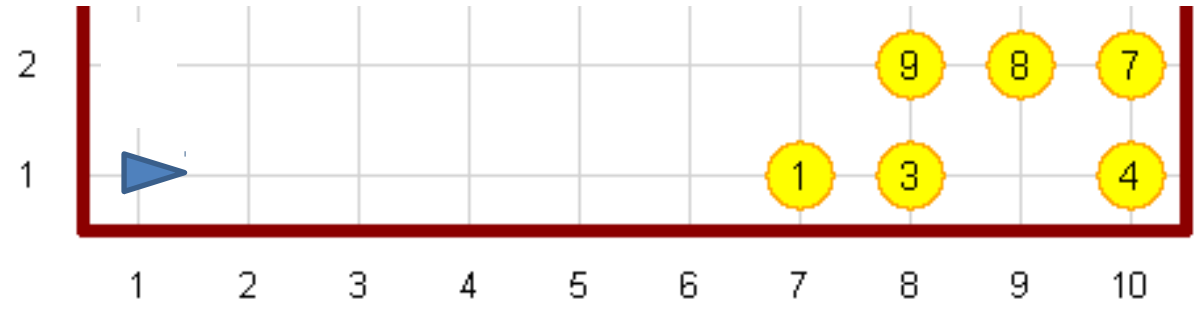


Pseudo code

1. While the front is not blocked.

1-1 Collect all beepers in a column and move to the next column if possible

2. Move back to the initial position.



How to collect all beeper in a non-empty column

```
def collect_ beepers_in_a_column():
```

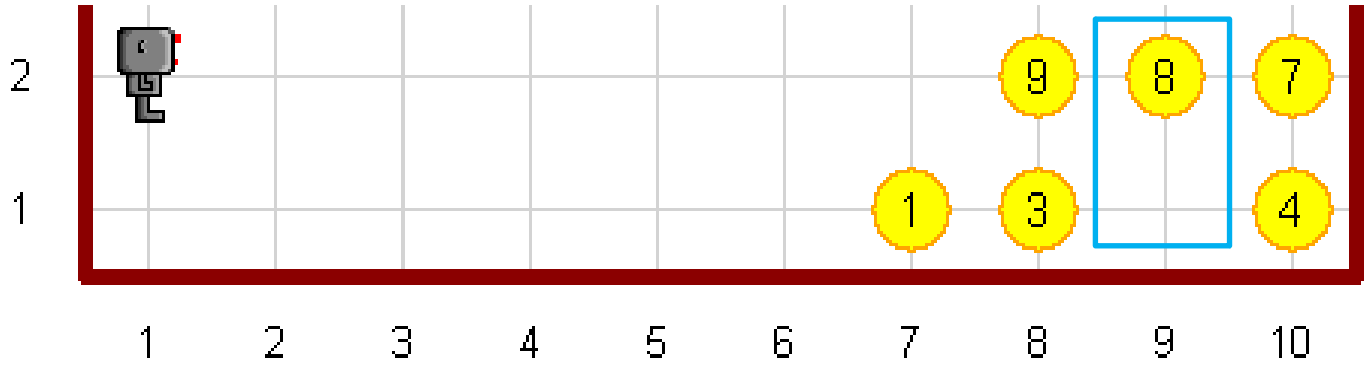
Move up to the second row.

Collect all beepers in the position if any.

Move down and add beepers.

Add all beepers.

Moving to the next column if possible.



How to collect all beepers in a place

```
def collect_beepers():
    while hubo.on_beeper():
        hubo.pick-beeper()
```

How to add all beepers in a place()

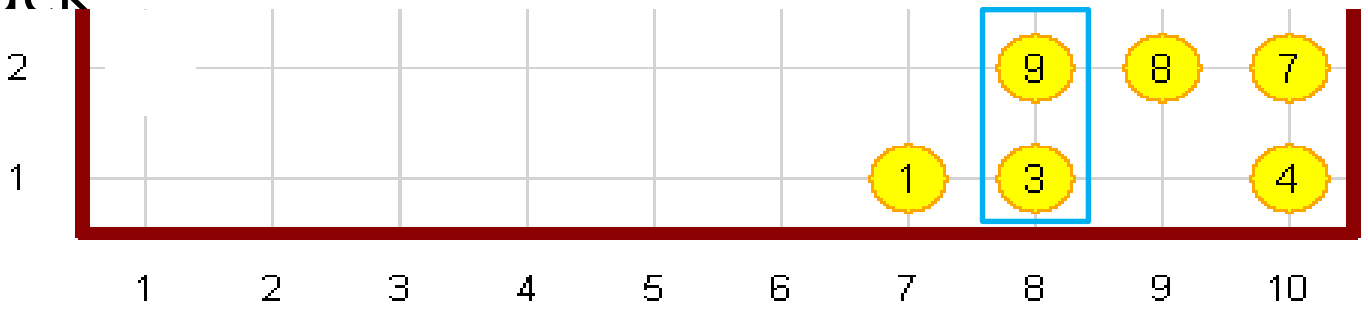
```
def add_beepers():
    while hubo.carries_beepers():
        hubo.drop_beeper()
```

PROBLEM20: The maximum number of beepers*

Modify your previous program so that it moves back to

the **column** with the **maximum number** of beepers. For

example, in the following figure, Hubo should move back to



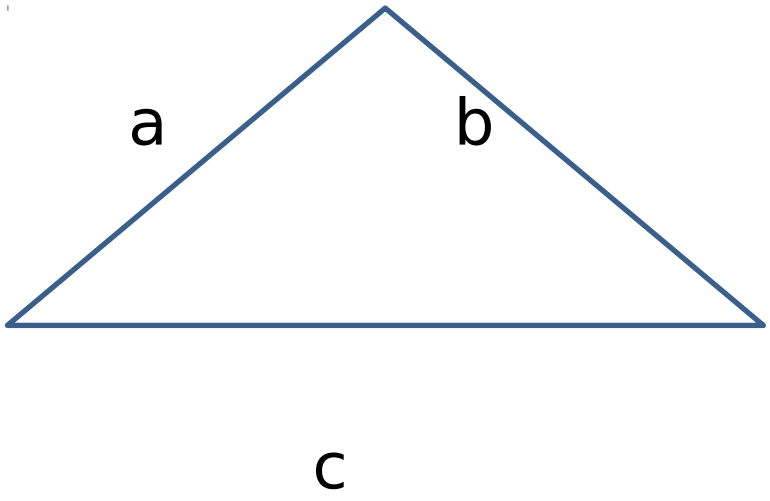
TRIANGULAR INEQUALITY

ASTU

PROBLEM 21: Triangular inequality*

Given three numbers a, b, and c, it is possible to form a triangle whose sides have length a, b, and c if and only if the **triangle inequality** holds. In other words, every side should be shorter than the sum of the oth er two sides. Write a program that checks if a triangl e can be formed with the three numbers. The three n umbers, a, b, c are provided by the user with a built i n function **raw_input** . As the output, the three num bers together with “True” or “False” should be printed depending on that the triangular inequality is hold tr ue or not

Triangular inequality



$a + b > c$, $b + c > a$, and $c + a > b$

Pseudo code

1. Input three numbers.
2. Check the triangular inequality for these numbers and report result.
3. Repeat Steps 1 and 2 until no more input is given.

A while-loop

Main program

```
while True:
    a, b, c = take_numbers()
    if check_inequality(a, b, c):
        print  a, b, c, “True”
    else:
        print a, b, c, “False”
    if raw_input(“Go for a more check ?”)  != “yes”:
        break
```

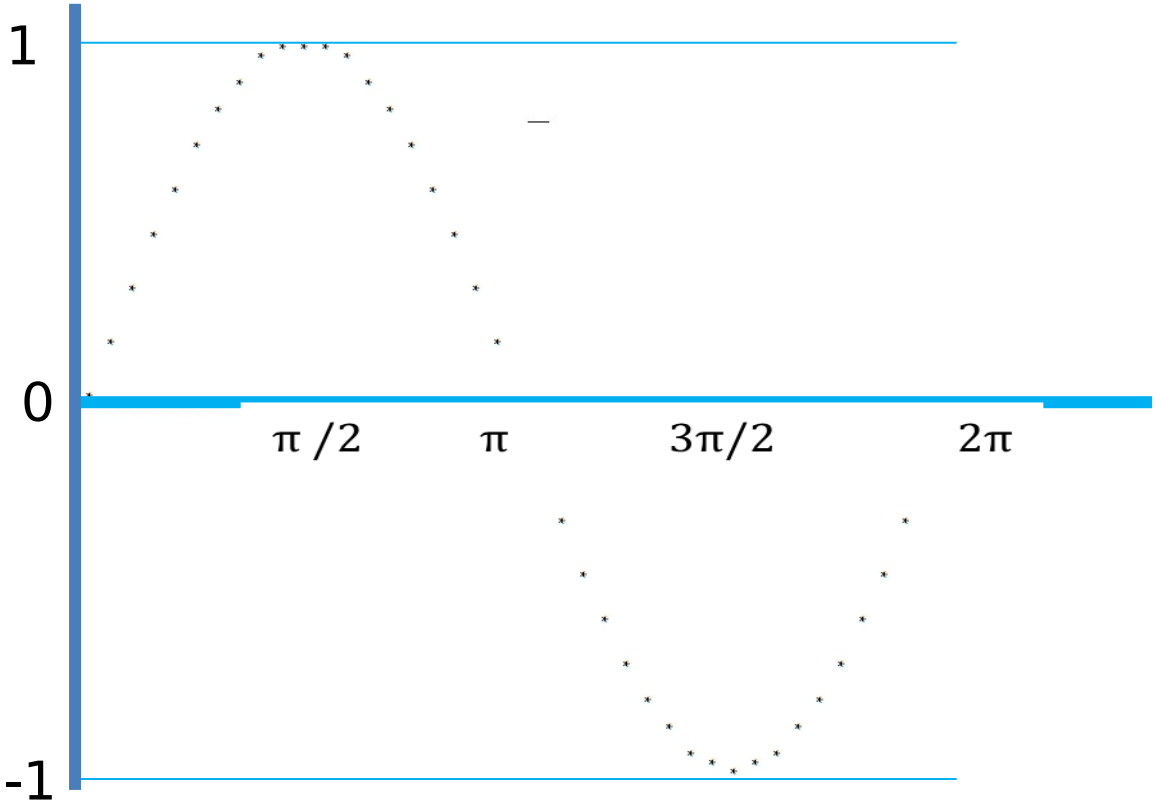
```
def take-numbers():
    Fill in this box.

    return x, y, z    # three number are assigned to x, y, z.


def check_inequality(x1, x2, x3):
    if x1 + x2 > x3 and x2 + x3 > x1 and x3 + x1 > x2:
        return True
    else:
        return False
```

DRAWING GRAPHS

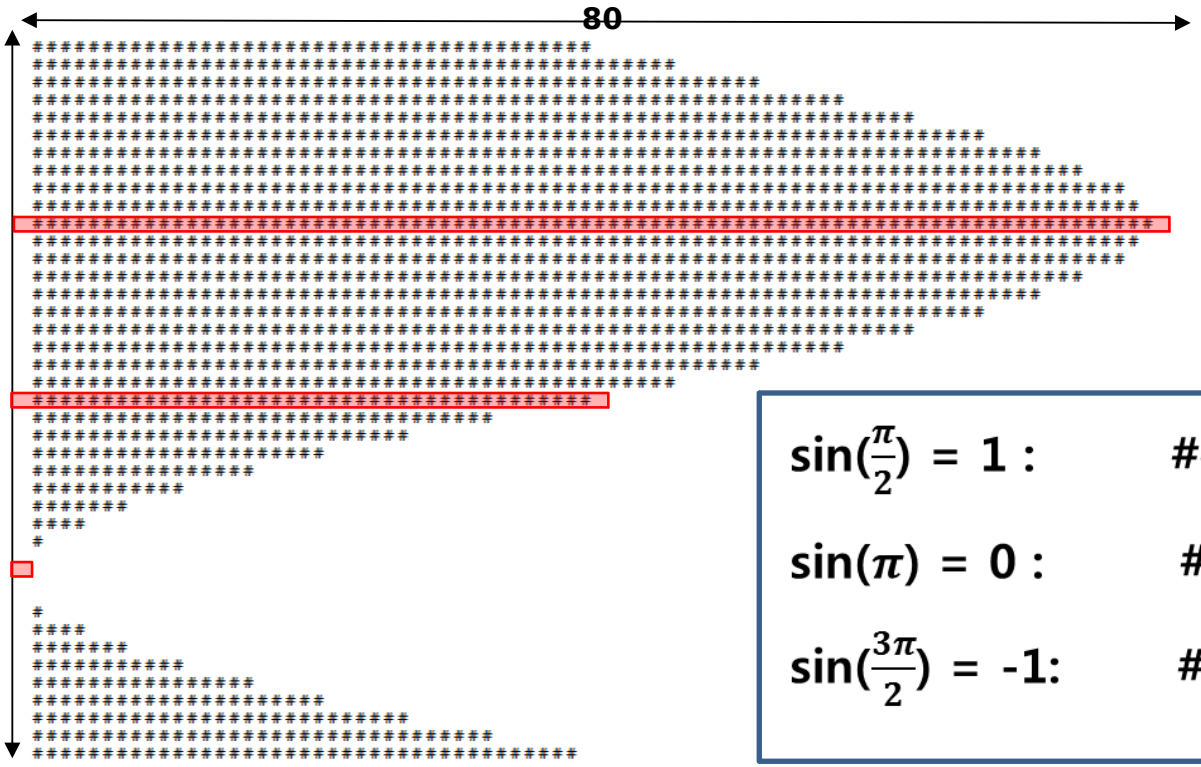
PROBLEM 22: Drawing sin curves*

A triangular function **sin(angle)** gives a value betwe
en -1 and 1, inclusively depending on angle in radians
as shown in the graph in the next slide. This graph sh
ows how sin(angle) changes as angle varies from 0 to
2π. You are asked to write a program that plots the cu
rve in three different forms: a bar graph and a point g
raph with axes.

Sine curve



1. Bar graph with #’s



$\sin(\frac{\pi}{2}) = 1$: ## · · · ##
80

$\sin(\pi) = 0$: ## · · · ##
40

$\sin(\frac{3\pi}{2}) = -1$: ## · · · ##
0

How to compute the number of #'s

$\sin(\frac{\pi}{2}) = 1$: ## · · · ##

80

$\sin(\pi) = 0$: ## · · · ##

40

$\sin(\frac{3\pi}{2}) = -1$: ## · · · ##

0

##

0

the number of #'s = sin(angle) * 40 + 40

Pseudo code

1. Change angle from 0 to 2 * π in k steps.
2. For each angle, compute the number of #'s and print the computed number of #'s

Employ for_loop to change the angle.

How to determine the number of steps k?

By trial and error! Try 40 steps.

Main program

```
import math
for i in range(41):
    angle = (2 * math.pi / 40) * float(i)
    compute_and_plot(angle)
```

Compute and plot

compute_and_plot(x):

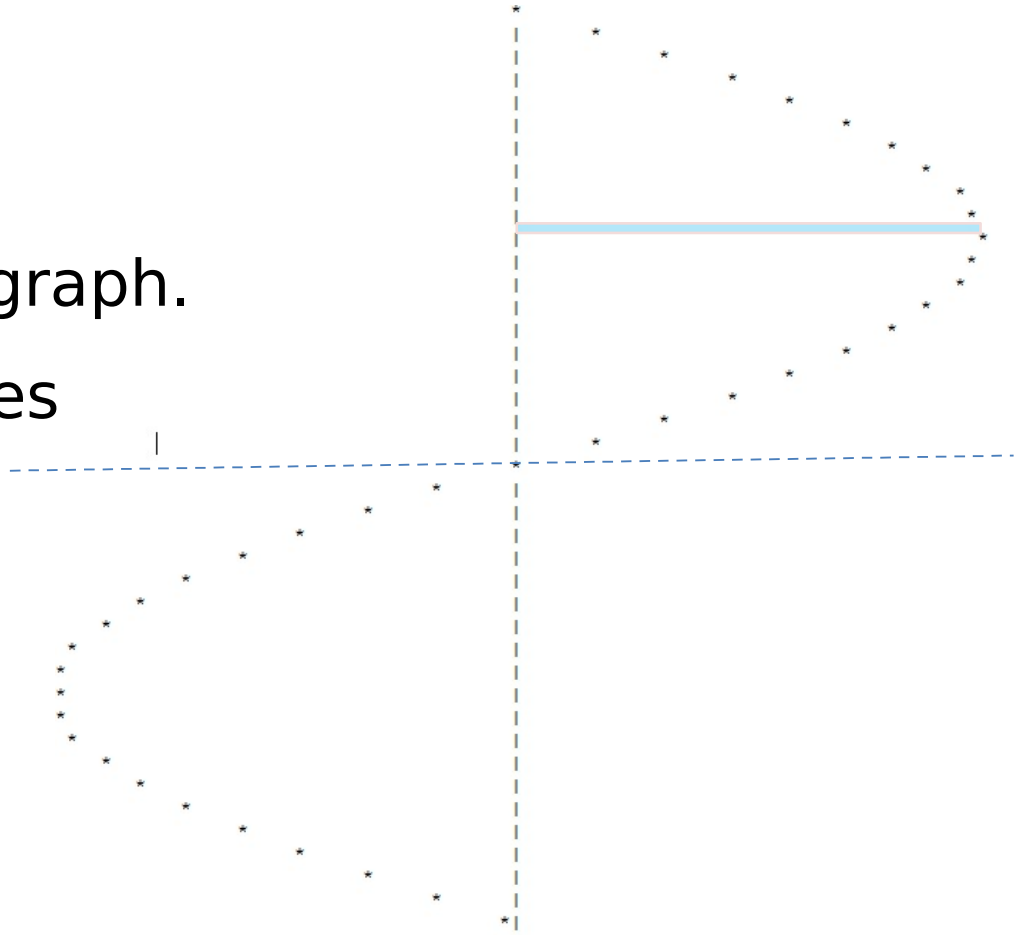
Compute the number of #’s for x.

Print the computed number of #’s.

2. Point graph with a axes

Pseudo code

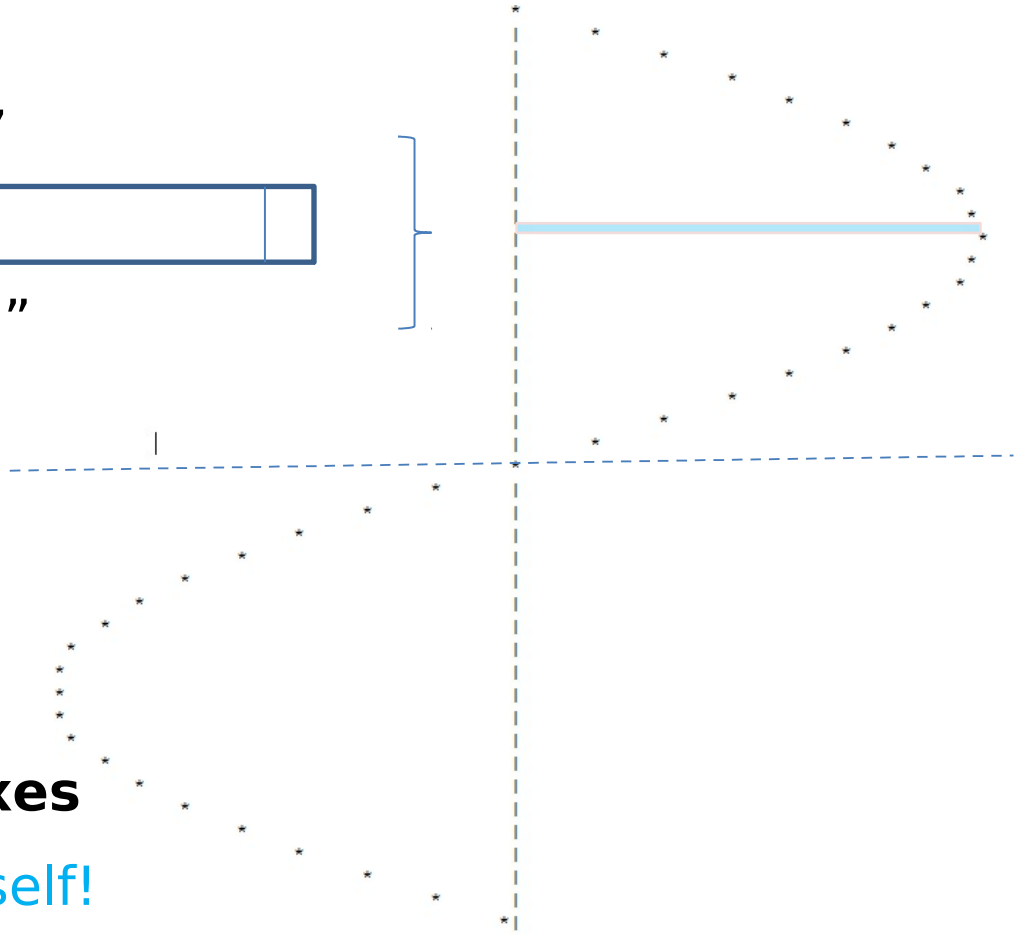
1. Plot a point graph.
2. Draw the axes



How to plot a point graph

39 blanks and “!!”

print 39 * “ “ + “!!”



How to draw the axes

Well, … do it by yourself!

PROBLEM S1: One way to improve your programming skill is to try to mimic a good programming style of other

people. We provide you with a printed copy of the

solution for PROBLEM 7. Please read this program carefully and re-implement it on a computer. You should also put your comments to show how well you understood the program.

PROBLEM S2: The code for this problem is a printed copy of the solution for PROBLEM 13. Please read this program carefully and re-implement it on a computer. You should also put your comments to show how well you understood the program.