University College Cork
Computer Science
Department

Programming Training

Main Points:

- Python Statements
- Problems with selections.

print()

print(value1, value2,…)

print(“var = ”, var1)

prints the text ‘var=‘ followed by the value of var

print(‘Here is a text’)

prints the text ‘Here is a text’

input()

input() returns a str value which must be converted

```
var = input()
    # reads a str value for var
var = float(input('Input var: '))
    # write the text 'Input var: ', then reads a str and converts that to float
var1, var2 = map(int, input('Input var1, var2').split())
    # reads two values, split them into a list and
    # then map the list values to var1, var2 as int
```

Some rules

Read an int variable var:

```
var = int(input('var = '))
```

Write a variable var:

```
print('var = ',var);
```

Write a text:

```
print("My Text \n");
```

Write a line of start:

```
printf("\n\n\n******* \n\n\n");
```

```
import math

def printFunction():

    print('\n\n**********\n\n')

    a = int(input('a = '))
    b = int(input('b = '))

    c = a+b

    print('c=', c)

    print('\n\n**********\n\n')
```

C Arithmetic

Arithmetic operators:

- +, -, *, /, **
- ++, -- for incrementing and decrementing.
- % is the remainder of the integer division.

Relational operators:

- <, >, <=, >=, ==, !=

Logical: or, and, not

There are several math functions in the math module

math.sqrt(),

math.sin(), math.cos(),

math.asin(), math.acos() etc.

How to solve a problem

1. Find the inputs and outputs of the problem.
2. Ask yourself if you have done something similar.
3. Identify the elements of the solution.
4. Take a numerical example and see the changes of the variables.
5. Write the code.

Inputs - Outputs

Inputs are the elements / variables to start from.

- Inputs must be read.
- Input can be arguments within functions.

Output(s) is represented by the problem result.

- Outputs must be printed.
- Outputs can be returned by a function.

The Process of Solving

1. Ask yourself if you have done something similar.
    1. Solve the problem mathematically.
2. Identify the elements of the solution:
    1. Break the problem into small blocks that you know how to.
    2. Write functions / find snippets of code for each block.
    3. Find the calculations / selections / repetitions
3. Write a pseudo-code in plain English.

Solving a triangle

- Given the sides *a, b, c* of a triangle then find the triangle angles *A, B, C* plus the area and the perimeter.

Inputs: a, b, c – double

Output: A, B, C, per and area.

Some mathematical equations

Cosine rule: $a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos(A)$

$$\cos(A) = \frac{b^2+c^2-a^2}{2 \cdot b \cdot c}$$

Solving a triangle – Pseudo-code

- Read a, b, c;

Calculate A = $acos\ \frac{b^2+c^2-a^2}{2 \cdot b \cdot c}$; and similar

Calculate per = a+b+c;

Calculate area = $\frac{b \cdot c \cdot \sin(A)}{2}$;

Print A, B, C, per and area

Solving a triangle

```
# import modules
import math

# define functions

def solveTriangle():

    # read inputs
    a = float(input('a = '))
    b = float(input('b = '))
    c = float(input('c = '))

    # calculate the angles in radians
    A = math.acos((b**2+c**2-a**2)/(2*b*c))
    B = math.acos((a**2+c**2-b**2)/(2*a*c))
    C = math.acos((a**2+b**2-c**2)/(2*a*b))

    # translate the angles in degree
    degreeA = A*180/math.pi
    degreeB = B*180/math.pi
    degreeC = C*180/math.pi

    # calculate the perimeter and area
    per = a+b+c
    area = b*c*math.sin(A)/2

    # print results
    print('Angles are: ', degreeA, degreeB, degreeC)
    print('Perimeter is: ', per)
    print('Area is: ', area)

# end solveTraingle
```

Comments:
The results are displayed with too many decimal.

round(value, decimals)

Python Statements.

Python statements give the flow of computation within a function.

1. Simple statements:
   - expression; - to evaluate a Python expression.
   - block - used to collect several statements in one block.
   - Jump:
     - break - jumps outside of repetitions.
     - return - gets outside of functions/routines
2. Selections
   - simple if - selects an alternative
   - complete if - selects from two alternatives

Python Blocks

Several lines which are identically indented form a block.

A block always starts after ':'

A block is usually required in functions and statements.

```
line1 of code
line2 of code
....
line of code :
    block line1
    block line2
    ....
    # end of block
```

INDENTATION MUST BE SIMILAR

Python tests - if statement.

Select one choice

```
if test :
    # block of if
    statement1;
    statement2;
    …
# end if
```

Choose between two choices

```
if test :
    # block of if
    statement1;
    statement2;
    …
else  :
    # block of else
    statement 1;
    statement 2;
    …
# end if
```

Choose between multiple choices

```
if test1 :
    # block of choice 1
    statement1;
    statement2;
    …
elif test2  :
    # block of  choice 2
    statement 1;
    statement 2;
    …
elif test3 :
   # block of  choice 2
    statement 1;
    statement 2;
    …
# end if
```

Minimum of two numbers.

Example: Find the maximum/minimum value of a,b

```
if a<b :
     max=b; min=a;
else
      max=a; min=b;
# endif
```

Testing Operators

Relational Operators:

>, >=, <, <=

== is equal to != is not equal to

Logical Operators:

or and not

Use ( ) to make complex test expressions

Max of 4 numbers.

Given for integer numbers a, b, c, d then

calculate the maximum value of them.

Inputs: a, b, c, d – int

Output: max – int.

How to do it?

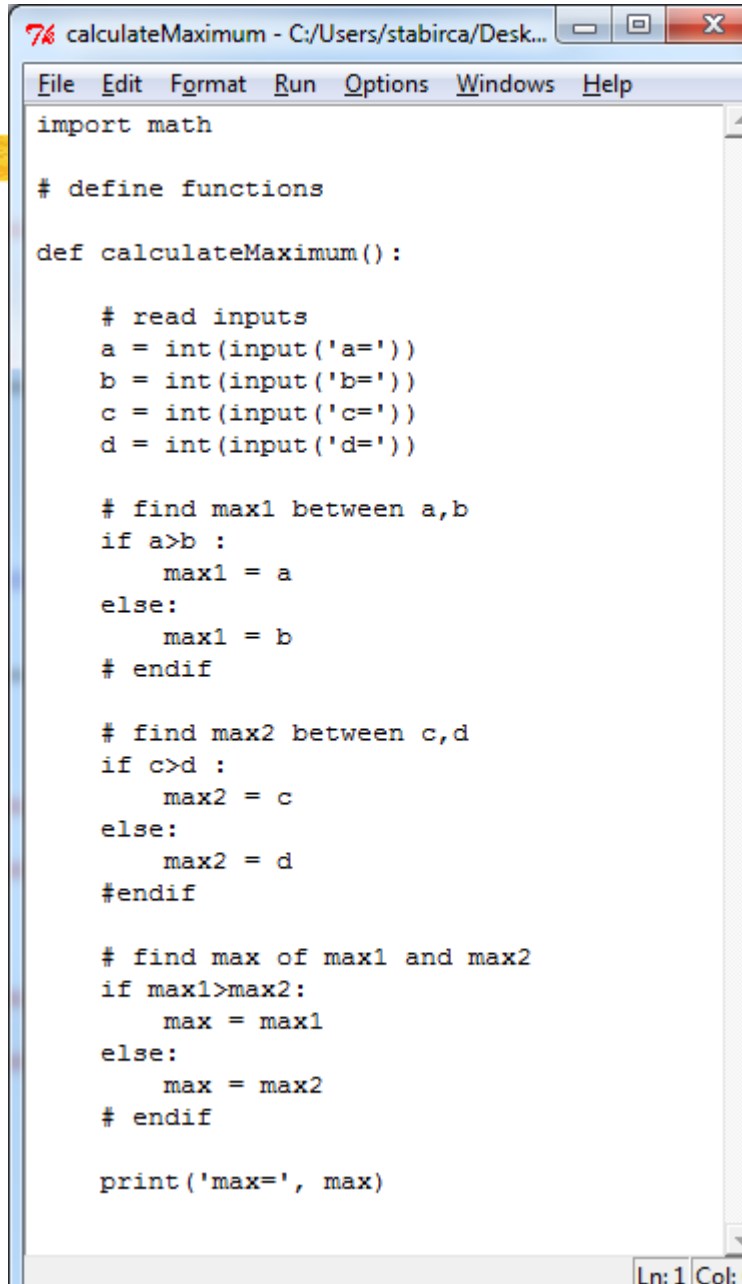
read a, b, c, d;

find max1 as the maximum of a and b

find max2 as the maximum of c and d

find max as the maximum of max1 and max2

write max



Test if three floats form a triangle.

Inputs: a, b, c – float

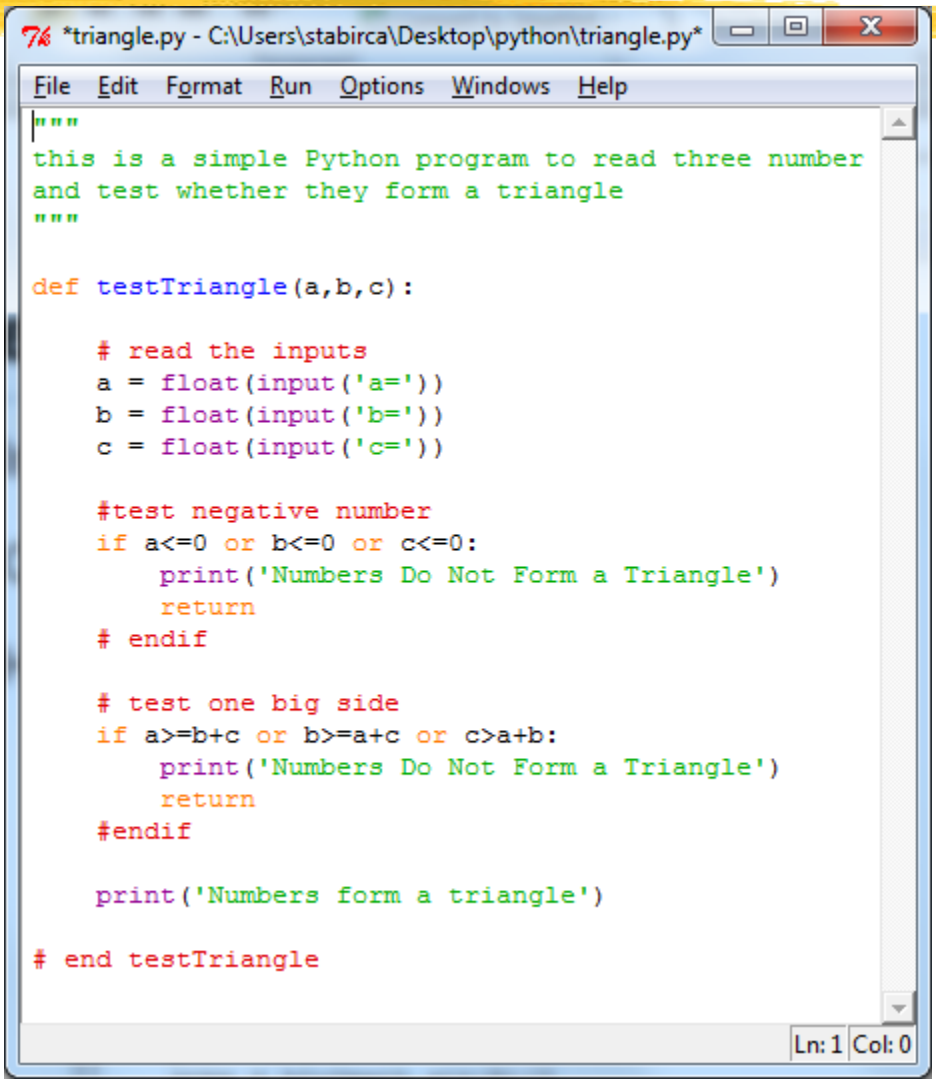
Output: answer.

How to do it?

read a, b, c;

test if at least one is negative

test if at least one number is too big



To do List

1. Read more about if from the e-tutorial.
2. Solve the HW problems.