

Contents

1	Basic Test Results	2
2	README	3
3	hello turtle.py	4
4	math print.py	5

1 Basic Test Results

```
1 Starting tests...
2 Wed Oct 21 21:39:32 IDT 2015
3 635f681c891e511c5afff62fd97288c5b384edd4 -
4
5
6 Archive: /tmp/bodek.vJC_w7/intro2cs/ex1/elinorperl/presubmission/submission
7   inflating: src/math_print.py
8   inflating: src/README
9   inflating: src/hello_turtle.py
10
11 Testing README...
12 Done testing README...
13
14 Running presubmit tests...
15 Wrong result, input: []:
16 expected: 1 4 9 16 25 36 49 64 81 100
17
18 actual:  1
19 4
20 9
21 16
22 25
23 36
24 49
25 64
26 81
27 100
28
29 result_code    ex1_t6    wrong    1
30 result_code    ex1      5      1
31 Done running presubmit tests
32
33 Tests completed
34
35 Additional notes:
36
37 Remember that hello_turtle.py was not tested at all.
38
39 If you are reading this by email, and have submitted multiple times,
40 your mail client may hide repeated errors as quoted text. Make sure
41 you do not miss them.
```

2 README

```
1  elinorperl
2  329577464
3  Elinor Perl
4  Yifat Natovich
5  Ofer Katz
6  Learning to use turtle by drawing a simple flower bed, as well as learning the "math" function and its basic uses.
7  Python 2 and 3's main differences are:
8  1. Unicode developement - improving the support.
9  2.Adjusting and refining the core language making it easier for new programers.
```

3 hello turtle.py

```
1  __author__ = 'elinorperl'
2  #####
3  # FILE: hello_turtle.py
4  # WRITER: Elinor Perl , elinorperl , 329577464
5  # EXCERISE: intro2cs ex1 2015-2016
6  # A simple program that demenstrates the use of defining functions using "Turtle"
7  #####
8
9  #I imported turtle - to draw a flower.
10 import turtle
11
12 #Stage 1 - Using the turtle function, I defined my own function to draw a petal, which I will later use to draw multiple petals
13 def draw_petal():
14     turtle.forward(30)
15     turtle.left(45)
16     turtle.forward(30)
17     turtle.left(135)
18     turtle.forward(30)
19     turtle.left(45)
20     turtle.forward(30)
21     turtle.left(135)
22
23 # Stage 2 - Using the turtle function, I used the earlier function for drawing petals and movements to form a flower.
24 def draw_flower():
25     turtle.right(45)
26     draw_petal()
27     turtle.right(90)
28     draw_petal()
29     turtle.right(90)
30     draw_petal()
31     turtle.right(90)
32     draw_petal()
33     turtle.right(135)
34     turtle.forward(150)
35
36 #Stage 3 - In order to make a series of flowers, the "turtle" must move to form the next one. This function draws one flower
37 def draw_flower_advanced():
38     draw_flower()
39     turtle.left(90)
40     turtle.up ()
41     turtle.forward(150)
42     turtle.left(90)
43     turtle.forward(150)
44     turtle.right(90)
45     turtle.down()
46
47 #Stage 4- Drawing the full bed of flowers. Once the "turtle" is in place to start and is able to start the flower, it can the
48 def draw_flower_bed():
49     turtle.up()
50     turtle.left(180)
51     turtle.forward(200)
52     turtle.right(180)
53     turtle.down()
54     draw_flower_advanced()
55     draw_flower_advanced()
56     draw_flower_advanced()
57     turtle.done()
58
59 draw_flower_bed()
```

4 math print.py

```
1 #####
2 # FILE: hello_print.py
3 # WRITER: Elinor Perl , elinorperl , 329577464
4 # EXCERSIZE: intro2cs ex1 2015-2016
5 # Working with math functions
6 #####
7
8 import math
9
10 def task_1():
11     print((1+(5)**(0.5))/2)
12     # I printed out the Golden Ratio by its direct equation
13 def task_2():
14     print(5**2)
15     # Directly through PyCharm's functions, I printed out 5 squared.
16 def task_3():
17     print ((4**2+5**2)**0.5)
18     # Using Pythagoras equation and directly using PyCharm's functions, I printed out what the hypotenuse equals.
19 def task_4():
20     print(math.pi)
21     #Using the math function, I printed out pi.
22
23 def task_5():
24     print(math.e)
25     #Using the math function, printed out e.
26 def task_6():
27     Area_Squared=1
28     while Area_Squared<=10:
29         print(Area_Squared**2)
30         Area_Squared=(Area_Squared)+1
31     #Useing "Loops", I printed each number from 1-10 squared.
```