

# Homework 2

## Problem 1 – Pythagorean Triples

Pseudo-code ( from

<http://mathforum.org/library/drmath/view/55811.html>)

- 1) Get integer input K
- 2) For m from 2 to K
  - a) For n from 1 to m
    - i) For d from 1 to K
      - (1)  $A = m^2 - n^2$
      - (2)  $B = 2mn$
      - (3)  $C = m^2 + n^2$
      - (4) Print( $d \cdot a$ ,  $d \cdot b$ ,  $d \cdot c$ )

## P1 – Code

```
# -*- coding: utf-8 -*-
"""
Homework 2, Problem 1: Pythagorean Triples
@author: daniel pettus
"""
# taken from http://mathforum.org/library/drmath/view/55811.html
max = 100
lst = []
for m in range(2,10):
    for n in range(1, m):
        for d in range(1, 10):
            a = (m**2 - n**2)
            b = 2*m*n
            c = (m**2 + n**2)
            print((d*a, d*b, d*c))
```

## Problem 2 – Duplicated Substrings

### P2.a Pseudo-code

#### Find dup str(s, n)

- 1) str\_lst = empty list
- 2) **FOR** i from zero **TO** len(s) - n + 1
  - a) **IF** substring **FROM** s[i] **TO** s[n+i-1] is in str\_lst
    - i) **RETURN** str\_lst
  - b) **ELSE** append substring to str\_lst
- 3) **RETURN** ""

### P2.c Pseudo-code

#### Find max dup(s)

- 1.) Dmax = empty string
- 2.) **FOR** n **FROM** 2 **TO** sqrt(len(s) / 2)
  - a. Ss = find\_dup\_str(s, n)
  - b. **IF** ss **IS EQUAL TO** empty string
    - i. Continue looping
  - c. **ELSE IF** ss > dmax
    - i. Dmax = ss
  - d. **RETURN**
    - i. dmax

## P2.b & P2.d Code

```
# Problem 2 - Duplicate strings
'''User enters a string, program looks for duplicates of size n'''
import math

def find_dup_str(s, n):
    '''Prints the substrings of size n across s'''
    str_lst = []
    for i in range(0, len(s) - n + 1):
        if s[i:i+n] in str_lst:
            return s[i:i+n]
        else:
            str_lst.append(s[i:i+n])
    return ""

def find_max_dup(s):
    '''half the len() is the biggest possible string'''
    dmax = ""
```

```

for n in range(2, math.ceil((len(s)/2))):
    ss = find_dup_str(s, n)
    if ss == "":
        continue
    elif ss > dmax:
        dmax = ss
return dmax

def main():
    '''Initialize Values before caring about input'''
    in_str = input("Enter text, then press <ENTER>: ")
    n = int(input("Enter an integer: "))
    print("s =", in_str, "\nn =", n)
    print(find_dup_str(in_str, n))
    s = 'abcgdefbczdgz'
    s2 = 'abcdefbcdgh'
    s3 = 'abcdefabcdghabcd'
    s4 = 'abchfurijdjfurabcpqweabc'
    print(find_max_dup(s))
    print(find_max_dup(s2))
    print(find_max_dup(s3))
    print(find_max_dup(s4))
main()

```

p2.py

1

# Problem 2 - Duplicate strings

2

'''User enters a string, program looks for du

3

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PS C:\Users\dp\Desktop\Python\h2> python.exe .\p2.py

Enter text, then press <ENTER>: abcdefbcdgh

Enter an integer: 2

a\_str = abcdefbcdgh

n\_int = 2

bc

PS C:\Users\dp\Desktop\Python\h2> python.exe .\p2.py

Enter text, then press <ENTER>: abcdefbcdgh

Enter an integer: 3

a\_str = abcdefbcdgh

n\_int = 3

bcd

PS C:\Users\dp\Desktop\Python\h2> python.exe .\p2.py

Enter text, then press <ENTER>: abcdefbcdgh

Enter an integer: 4

a\_str = abcdefbcdgh

n\_int = 4

shown for debugging

PS C:\Users\dp\Desktop\Python\h2>

PS C:\Users\dp\Desktop\Python\h2> python.exe .\p2.py

Enter text, then press <ENTER>: abchfurijdjfurabcpqweabc

Enter an integer: 3

s = abchfurijdjfurabcpqweabc

n = 3

fur

bc

bcd

abcd

PS C:\Users\dp\Desktop\Python\h2> python.exe .\p2.py

Enter text, then press <ENTER>: abchfurijdjfurabcpqweabc

Enter an integer: 3

s = abchfurijdjfurabcpqweabc

n = 3

fur

bc

bcd

abcd

fur

PS C:\Users\dp\Desktop\Python\h2>

## Problem 3 – Function Visualization

```
# -*- coding: utf-8 -*-
'''Evaluates user's equation, then prints a table and graph of results'''
import math
import pylab

def main():
    '''Takes input, computes and displays results'''
    fun_str = input('Enter function with variable x: ')
    ns = float(input('Enter number of samples: '))
    xmin = float(input('Enter xmin: '))
    xmax = float(input('Enter xmax: '))
    x = xmin
    xs = []
    ys = []
    dx = (xmax - xmin) / ns
    while x <= xmax:
        xs.append(x)
        y = eval(fun_str)
        ys.append(y)
        x += dx
    print(xs)
    print(ys)
    print('{:>20s}{:>20s}'.format('x', 'y'))
    print('-' * 40)
    for i in range(len(xs)):
        print('{:>20.4f}{:>20.4f}'.format(xs[i], ys[i]))

    pylab.title(fun_str)
    pylab.plot(xs, ys, '-bo')
    pylab.xlim(xmin, xmax)
    pylab.ylim(min(ys), max(ys))
    pylab.show()
main()
```

```
In [1]: run p3_v1.py
```

```
Enter function with variable x: x**2
```

```
Enter number of samples: 10
```

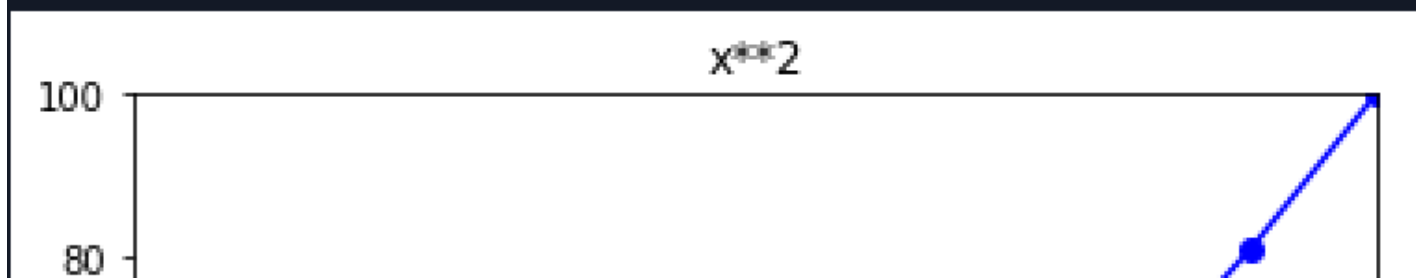
```
Enter xmin: 0
```

```
Enter xmax: 10
```

```
[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0]
```

```
[0.0, 1.0, 4.0, 9.0, 16.0, 25.0, 36.0, 49.0, 64.0, 81.0, 100.0]
```

x	y
0.0000	0.0000
1.0000	1.0000
2.0000	4.0000
3.0000	9.0000
4.0000	16.0000
5.0000	25.0000
6.0000	36.0000
7.0000	49.0000
8.0000	64.0000
9.0000	81.0000
10.0000	100.0000



9.0000  
10.0000

81.0000  
100.0000

$x^2$

