國立臺灣大學

Problem Set — For

Computer Programming (EE3031), F21

Jiun-Lang Huang

EE Department, National Taiwan University

Problem F01: Square Root

One way of computing square roots is the Newton's method. Suppose that you want to compute the square root of x. Let a_i be your i-th guess. a_{i+1}, a better guess, can be obtained by $a_{i+1} = \dfrac{a_i + \dfrac{x}{a_i}}{2}$.

For example, let $x = 5$ and the initial guess of $\sqrt{x}$ be $\dfrac{x}{2}$, i.e., 2.5. We have $a_1 = \dfrac{a_0 + \dfrac{x}{a_0}}{2} = \dfrac{2.5 + \dfrac{5}{2.5}}{2} = 2.25$. By repeating the process, we can obtain a good approximation of $\sqrt{5}$.

- Write a program `square_root.py` that asks the user to enter a positive integer, x. Then, apply the Newton's method (with $a_0 = \frac{x}{2}$) and shows the the first ten guesses.

- Example IO:

```
fin2017 > python3 square_root.py
enter a positive integer: 5
guess 0: 2.50000000
guess 1: 2.25000000
guess 2: 2.23611111
guess 3: 2.23606798
guess 4: 2.23606798
guess 5: 2.23606798
guess 6: 2.23606798
guess 7: 2.23606798
guess 8: 2.23606798
guess 9: 2.23606798
fin2017 >
```

Problem F02: Draw Rectangle v1

- Write *draw_rectangle_1.py* which draws a rectangle of user specified width and height. Fill the rectangle with '*'.

```
fin2017 > python3 draw_rectangle_1.py
enter the width: 8
enter the height: 4
********
********
********
********
fin2017 >
```

Problem F03: Draw Rectangle v2

- Write *draw_rectangle_2.py* which draws, with '*', a hollow rectangle of user specified width and height.
  - *width* and *height* are both positive integers.
  - If *width* = 1/*height* = 1, the hollow rectangle degenerates into a vertical/horizontal line.

```
fin2017 > python3 F03_draw_rectangle_2.py
enter the width: 4
enter the height: 6
****
*  *
*  *
*  *
*  *
****
fin2017 >
```

```
fin2017 > python3 F03_draw_rectangle_2.py
enter the width: 1
enter the height: 1
*
fin2017 >
```

```
fin2017 > python3 F03_draw_rectangle_2.py
enter the width: 2
enter the height: 3
**
**
**
fin2017 >
```

```
fin2017 > python3 F03_draw_rectangle_2.py
enter the width: 6
enter the height: 2
******
******
fin2017 >
```

```
fin2017 > python3 F03_draw_rectangle_2.py
enter the width: 1
enter the height: 3
*
*
*
fin2017 >
```

```
fin2017 > python3 F03_draw_rectangle_2.py
enter the width: 4
enter the height: 1
****
fin2017 >
```

Problem F04: Draw Isosceles Right Triangle

- Write *draw_right_triangle.py* which draws a isosceles right triangle (等腰直角三角形) with user specified height. Fill the rectangle with '*'.

```
fin2017 > python3 F04_draw_right_triangle.py
enter the height: 1
*
fin2017 > python3 F04_draw_right_triangle.py
enter the height: 5
*
**
***
****
*****
fin2017 >
```

Problem F05: Draw Rectangle v3

- Write *draw_rectangle_3.py* which draws a rectangle of user specified width and height. Fill the rectangle with integers starting from 0.

```
fin2017 > python3 F05_draw_rectangle_3.py
enter the width: 3
enter the height: 3
0 1 2
3 4 5
6 7 8
fin2017 > python3 F05_draw_rectangle_3.py
enter the width: 5
enter the height: 7
 0  1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 32 33 34
fin2017 >
```

```
fin2017 > python3 F05_draw_rectangle_v3.py
enter the width: 10
enter the height: 15
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14
 15  16  17  18  19  20  21  22  23  24  25  26  27  28  29
 30  31  32  33  34  35  36  37  38  39  40  41  42  43  44
 45  46  47  48  49  50  51  52  53  54  55  56  57  58  59
 60  61  62  63  64  65  66  67  68  69  70  71  72  73  74
 75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
 90  91  92  93  94  95  96  97  98  99 100 101 102 103 104
105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134
135 136 137 138 139 140 141 142 143 144 145 146 147 148 149
fin2017 >
```

Problem 06: Digit Cube Sum Integer

- There exist exactly four integers greater than one which are the sums of the cubes of their digits. For example, one of them is 153 as $1^3 + 5^3 + 3^3 = 153$.
- Write a program *digit_cube_sum_int.py* that finds and prints out the four numbers.
- Hint: All four numbers are less than 500.