KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

Homework # 5

13006107 Introduction to Computers and Programming

Software Engineering Program

Faculty of Engineering, KMITL

By

64011655 Teerapat Wattanamanont

Name-Surname: Teerapat Wattanamanont ID: 64011655

Introduction to Computers and Programming, SE Program

Homework #5

6th September 2021

1. The Problem

You need to find the square root of a number, but unfortunately you are a poor Babylonian back in 2000 B. C. without a calculator. Fortunately, your buddy down the road has come up with a cute little algorithm (he discovered it while minding his sheep, sharp guy!) that gets a pretty good approximation of a square root. Here is the algorithm:

1. Prompt the user for a number *n*, of which you will find its square root
2. Make an initial *guess* of the square root (n/2 is a good first guess).
3. Create a new float variable *temp*
4. Set *temp* = *n*/*guess*
5. Update *guess* to have the value *guess*=(*guess* + *temp*)/2

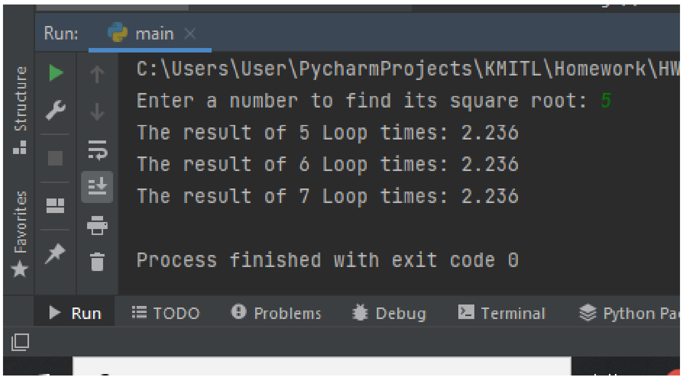
Repeat steps 4 and 5 to get ever closer to the real answer.

Your Task

Your task is to implement the above little algorithm (which is more commonly known as Newton's method. It is controversial whether ancient Babylonian's actually knew this algorithm). **To make it simpler**, iterate 5 times the step 4-5 calculation that should be sufficient for a square root approximation (however, for the approximation to be more accurate the iteration could be done more than 5 times).

Write a Python program to iterate the step 4-5 calculation with 5, 6, and 7 time respectively in order to compare the approximation results and report your answers in the three decimal points of accuracy.

```python
# TODO 1
user_input = int(input("Enter a number to find its square root: "))
guess = user_input / 2

for i in range(5):
    temp = user_input / guess
    guess = (guess + temp) / 2
for i in range(6):
    temp2 = user_input / guess
    guess = (guess + temp) / 2
for i in range(7):
    temp3 = user_input / guess
    guess = (guess + temp) / 2

new_temp = float("{0:.3f}".format(temp))
new_temp2 = float("{0:.3f}".format(temp2))
new_temp3 = float("{0:.3f}".format(temp3))

print(f"The result of 5 Loop times: {new_temp}")
print(f"The result of 6 Loop times: {new_temp2}")
print(f"The result of 7 Loop times: {new_temp3}")
```



```
C:\Users\User\PycharmProjects\KMITL\Homework\HW
Enter a number to find its square root: 5
The result of 5 Loop times: 2.236
The result of 6 Loop times: 2.236
The result of 7 Loop times: 2.236

Process finished with exit code 0
```

2. Write a Python program using the turtle module and **while** loops to print out the calendar of 12 months of year 2021 in the following format.

Month#1						
Su	Mo	Tu	We	Th	Fr	Sa
					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						

Month#2						
Su	Mo	Tu	We	Th	Fr	Sa
	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						

……

Month#12						
Su	Mo	Tu	We	Th	Fr	Sa
			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	

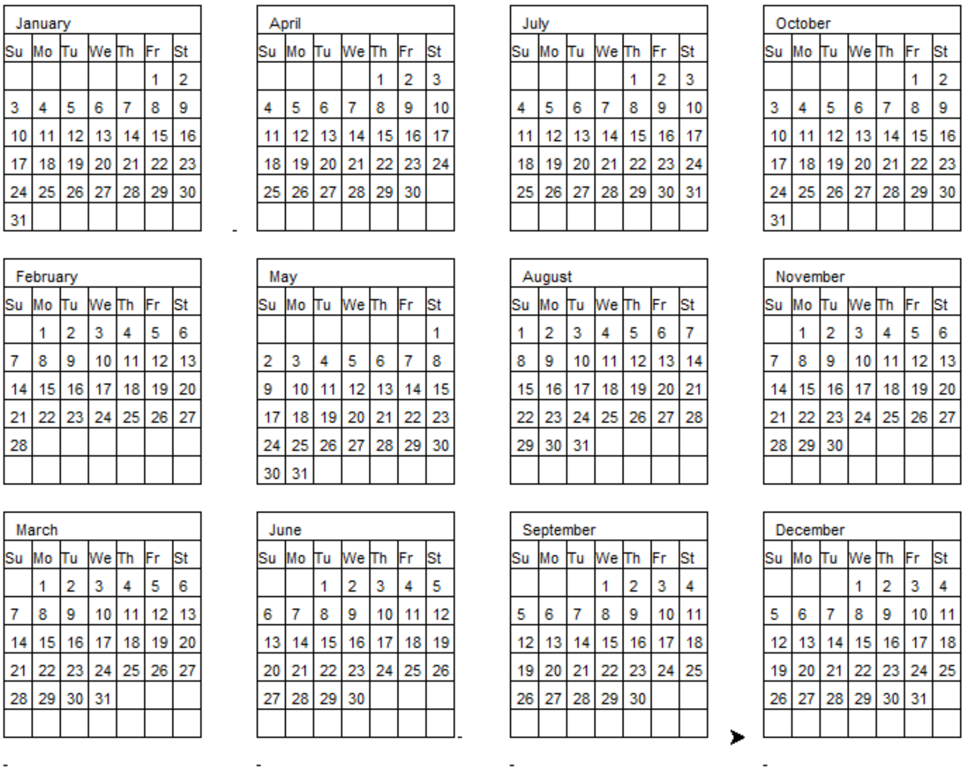
```
 # TODO 2
from turtle import Turtle, Screen

weekdays = ["", "Su", "Mo", "Tu", "We", "Th", "Fr", "St"]
dates1 = ["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"]

month = 1
square_size = 20
day = 1
dates = 1

tim = Turtle()
screen = Screen()
tim.speed("fastest")


def go_back():
    tim.backward(square_size + 3)
    tim.backward(square_size * 6)


def go_back2():
    tim.setheading(270)
    tim.penup()
    tim.forward(square_size)
    tim.pendown()
    tim.setheading(0)
    tim.forward(3)
    tim.penup()
```

```
def go_back3():
    go_back()
    go_back2()


def write_date(start_day, end_date):
    a = start_day
    while (a <= end_date):
        tim.write(str(a))
        a += 1
        tim.forward(square_size)


def draw_box(month_name):
    for i in range(2):
        tim.forward(square_size * 7)
        tim.right(90)
        tim.forward(20)
        tim.right(90)

    # Write month name
    tim.setheading(90)
    tim.backward(square_size)
    tim.setheading(0)
    tim.forward(10)
    tim.write(month_name)
    tim.backward(10)
    day = 1

    # Draw squares and write day names
    day = 1
    while day < 8:
        for i in range(2):
            tim.forward(square_size)
            tim.right(90)
        tim.forward(square_size - 2)
        tim.right(90)
        tim.write(weekdays[day])
        tim.setheading(180)
        tim.forward(2)
        tim.setheading(90)
        tim.forward(square_size)
        tim.setheading(0)
        tim.forward(square_size)
        day += 1

    tim.backward(square_size * 7)
    tim.right(90)
    tim.forward(square_size)
    tim.left(90)
    for i in range(6):
        for j in range(7):
            for k in range(2):
                tim.forward(square_size)
                tim.right(90)
                tim.forward(square_size)
                tim.right(90)

            tim.forward(square_size)
        tim.setheading(270)
        tim.forward(square_size)
        tim.setheading(0)
        tim.backward(square_size * 7)

    go_back2()


def jan_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size * 5)
    tim.forward(3)
    tim.write(dates1[0])
    tim.forward(square_size)
    tim.write(dates1[1])
    tim.backward(3)
    tim.backward(square_size * 6)
    tim.right(90)
    tim.forward(square_size)
    tim.left(90)
    tim.forward(3)
    write_date(3, 9)
    go_back()
    go_back2()
    write_date(10, 16)
    go_back3()
    write_date(17, 23)
    go_back3()
    write_date(24, 30)
    go_back()
    tim.right(90)
    tim.forward(square_size)
    tim.left(90)
    tim.forward(3)
    tim.write(dates1[30])


def feb_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size)
    tim.forward(3)
    write_date(1, 6)
    go_back3()
    write_date(7, 13)
    go_back3()
    write_date(14, 20)
    go_back3()
    write_date(21, 27)
    go_back3()
    tim.write("28")


def march_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size)
    tim.forward(3)
    write_date(1, 6)
    go_back3()
    write_date(7, 13)
    go_back3()
    write_date(14, 20)
    go_back3()
    write_date(21, 27)
    go_back3()
    write_date(28, 31)
    go_back3()


def april_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size * 4)
    tim.forward(3)
    write_date(1, 3)
    go_back3()
    write_date(4, 10)
    go_back3()
    write_date(11, 17)
    go_back3()
    write_date(18, 24)
    go_back3()
    write_date(25, 30)
    go_back3()


def may_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size * 6)
    tim.forward(3)
    tim.write(dates1[0])
    tim.forward(square_size - 3)
    tim.backward(square_size * 7)
    tim.right(90)
    tim.forward(square_size)
    tim.left(90)
    tim.forward(3)
    write_date(2, 8)
    go_back3()
    write_date(9, 15)
    go_back3()
    write_date(17, 23)
    go_back3()
    write_date(24, 30)
    go_back3()

    tim.write(dates1[29])
    tim.forward(square_size)
    tim.write(dates1[30])
```

```
def jun_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size * 2)
    tim.forward(3)
    write_date(1, 5)
    go_back3()
    write_date(6, 12)
    go_back3()
    write_date(13, 19)
    go_back3()
    write_date(20, 26)
    go_back3()
    write_date(27, 30)
    go_back3()


def jul_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size * 4)
    tim.forward(3)
    write_date(1, 3)
    go_back3()
    write_date(4, 10)
    go_back3()
    write_date(11, 17)
    go_back3()
    write_date(18, 24)
    go_back3()
    write_date(25, 31)
    go_back3()


def aug_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(3)
    write_date(1, 7)
    go_back3()
    write_date(8, 14)
    go_back3()
    write_date(15, 21)
    go_back3()
    write_date(22, 28)
    go_back3()
    # # Write date 28
    write_date(29, 31)
    go_back3()


def sep_cal():
    tim.left(90)
```

```python
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size * 3)
    tim.forward(3)
    write_date(1, 4)
    go_back3()
    write_date(5, 11)
    go_back3()
    write_date(12, 18)
    go_back3()
    write_date(19, 25)
    go_back3()
    write_date(26, 30)
    go_back3()


def oct_cal():
    jan_cal()


def nov_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size)
    tim.forward(3)
    write_date(1, 6)
    go_back3()
    write_date(7, 13)
    go_back3()
    write_date(14, 20)
    go_back3()
    write_date(21, 27)
    go_back3()
    write_date(28, 30)
    go_back3()


def dec_cal():
    tim.left(90)
    tim.penup()
    tim.forward(square_size * 6)
    tim.right(90)
    tim.forward(square_size * 3)
    tim.forward(3)
    write_date(1, 4)
    go_back3()
    write_date(5, 11)
    go_back3()
    write_date(12, 18)
    go_back3()
    write_date(19, 25)
    go_back3()
    write_date(26, 31)
    go_back3()


# JAN
tim.penup()
tim.goto(-450, 300)
tim.pendown()
draw_box("January")
jan_cal()

tim.penup()
tim.goto(-450, 120)
tim.pendown()

# FEB
draw_box("February")
feb_cal()

tim.penup()
tim.goto(-450, -60)
tim.pendown()

# MAR
draw_box("March")
march_cal()

tim.penup()
tim.goto(-270, 300)
tim.pendown()

# APR
draw_box("April")
april_cal()

tim.penup()
tim.goto(-270, 120)
tim.pendown()

# MAY
draw_box("May")
may_cal()

tim.penup()
tim.goto(-270, -60)
tim.pendown()

# JUN
draw_box("June")
jun_cal()

tim.penup()
tim.goto(-90, 300)
tim.pendown()

# JUL
draw_box("July")
jul_cal()

tim.penup()
tim.goto(-90, 120)
tim.pendown()

# AUG
draw_box("August")
aug_cal()
```

```
tim.penup()
tim.goto(-90, -60)
tim.pendown()

# SEP
draw_box("September")
sep_cal()

# October
tim.penup()
tim.goto(90, 300)
tim.pendown()

draw_box("October")
oct_cal()

# November
tim.penup()
tim.goto(90, 120)
tim.pendown()

draw_box("November")
nov_cal()

# December
tim.penup()
tim.goto(90, -60)
tim.pendown()

draw_box("December")
dec_cal()

screen.exitonclick()
```



January

Su	Mo	Tu	We	Th	Fr	St
					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						

February

Su	Mo	Tu	We	Th	Fr	St
	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						

March

Su	Mo	Tu	We	Th	Fr	St
	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			

April

Su	Mo	Tu	We	Th	Fr	St
				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	

May

Su	Mo	Tu	We	Th	Fr	St
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
17	18	19	20	21	22	23
24	25	26	27	28	29	30
30	31					

June

Su	Mo	Tu	We	Th	Fr	St
		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			

July

Su	Mo	Tu	We	Th	Fr	St
				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

August

Su	Mo	Tu	We	Th	Fr	St
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				

September

Su	Mo	Tu	We	Th	Fr	St
			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		

October

Su	Mo	Tu	We	Th	Fr	St
					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						

November

Su	Mo	Tu	We	Th	Fr	St
	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				

December

Su	Mo	Tu	We	Th	Fr	St
			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	

3. Write a Python program that prompts the user to enter any integer, greater than or equal to 1, and the program displays the output with the pattern like the following **examples**:

Input: 1

```
*
```

Input: 3

```
*
**
***
**
*
**
*
*
```

Input: 5

```
*
**
***
****
*****
****
***
**
*
**
***
****
***
**
*
**
***
**
*
**
*
*
```

```
# TODO 3
user_input = int(input("Enter an integer: "))

if user_input < 1:
    print("Error\nPlease enter an integer that is greater or equal to 1")
elif user_input == 1:
    print("*")
else:
    for i in range(1, user_input):
        for j in range(1, user_input):
            for k in range(1, j + 2):
                print("*", end="")
            print()
        for i in range(user_input, 1, -1):
            for j in range(0, user_input - 1):
                print("*", end="")
            print()
        user_input -= 1
    print("*")
```