Scientific Computing with Python Lab

4th Session(Feb 6th)

Kaheon Kim

Today

we are going to talk about

- #Review_Homework_2 : while loop & mathematical concept
- #FAQ
- #function
- #Other_domain

FAQ

1. Are there any built in package for the tasks in homeworks?

-> A : Yes! Numpy, Pandas, Tensorflow, Pytorch, Scipy …

2. Why we don't use that fancy packages?

-> A: Basically, it's ACMS course, centered on interpreting the problem and moving it into the implementation.

Homework 2

While loop

While loop: iterate **under certain condition is satisfied**

```
while (condition) :
    operation
```

```
while (True) :
    operation
    if (stop_condition):
        break
```

→ so we have to make iterative condition

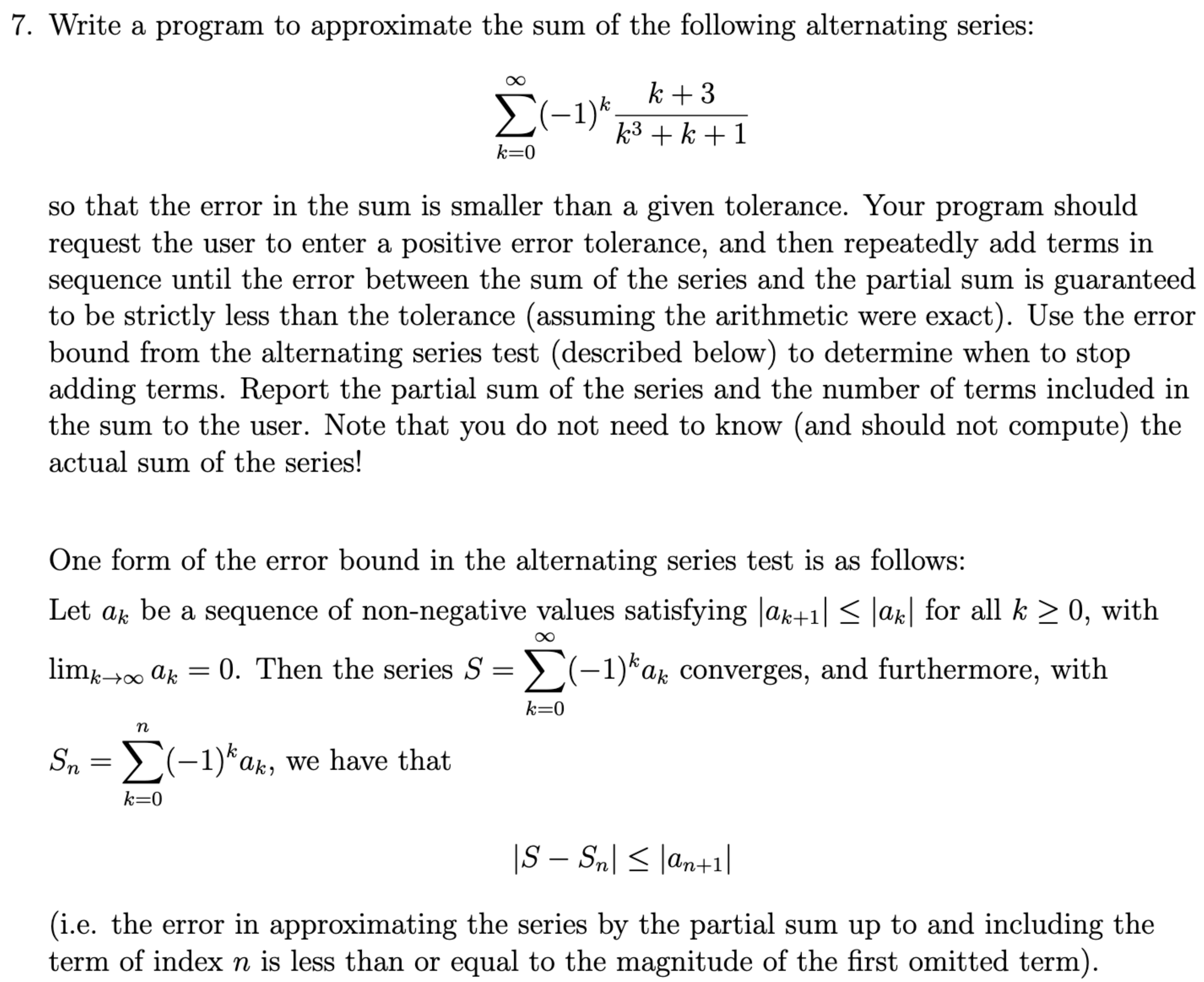
Homework 2

While loop : compare with for statement

	for statement	**while statement**
index	set of indices	we have to update for each loop by ourselves
stopping condition	if you used up all the indices	if it starts to disobey the iterative condition

Homework 2

Question 7



7. Write a program to approximate the sum of the following alternating series:

$$\sum_{k=0}^{\infty}(-1)^k \frac{k+3}{k^3+k+1}$$

so that the error in the sum is smaller than a given tolerance. Your program should request the user to enter a positive error tolerance, and then repeatedly add terms in sequence until the error between the sum of the series and the partial sum is guaranteed to be strictly less than the tolerance (assuming the arithmetic were exact). Use the error bound from the alternating series test (described below) to determine when to stop adding terms. Report the partial sum of the series and the number of terms included in the sum to the user. Note that you do not need to know (and should not compute) the actual sum of the series!

One form of the error bound in the alternating series test is as follows:

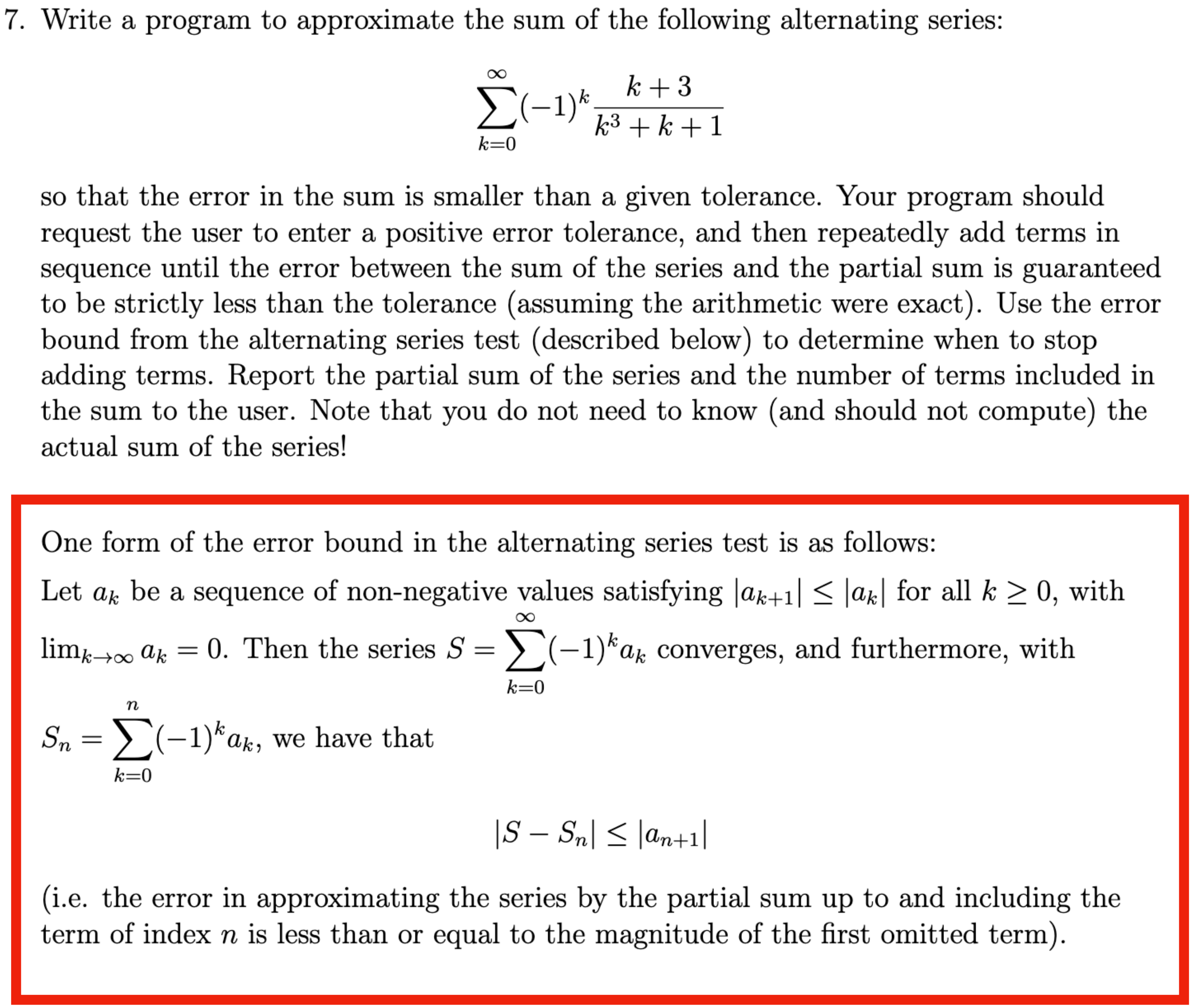
Let a_k be a sequence of non-negative values satisfying $|a_{k+1}| \leq |a_k|$ for all $k \geq 0$, with $\lim_{k\to\infty} a_k = 0$. Then the series $S = \sum_{k=0}^{\infty}(-1)^k a_k$ converges, and furthermore, with $S_n = \sum_{k=0}^{n}(-1)^k a_k$, we have that

$$|S - S_n| \leq |a_{n+1}|$$

(i.e. the error in approximating the series by the partial sum up to and including the term of index n is less than or equal to the magnitude of the first omitted term).

Homework 2

Question 7



7. Write a program to approximate the sum of the following alternating series:

$$\sum_{k=0}^{\infty}(-1)^k \frac{k+3}{k^3+k+1}$$

so that the error in the sum is smaller than a given tolerance. Your program should request the user to enter a positive error tolerance, and then repeatedly add terms in sequence until the error between the sum of the series and the partial sum is guaranteed to be strictly less than the tolerance (assuming the arithmetic were exact). Use the error bound from the alternating series test (described below) to determine when to stop adding terms. Report the partial sum of the series and the number of terms included in the sum to the user. Note that you do not need to know (and should not compute) the actual sum of the series!

One form of the error bound in the alternating series test is as follows:

Let a_k be a sequence of non-negative values satisfying $|a_{k+1}| \leq |a_k|$ for all $k \geq 0$, with $\lim_{k\to\infty} a_k = 0$. Then the series $S = \sum_{k=0}^{\infty}(-1)^k a_k$ converges, and furthermore, with $S_n = \sum_{k=0}^{n}(-1)^k a_k$, we have that

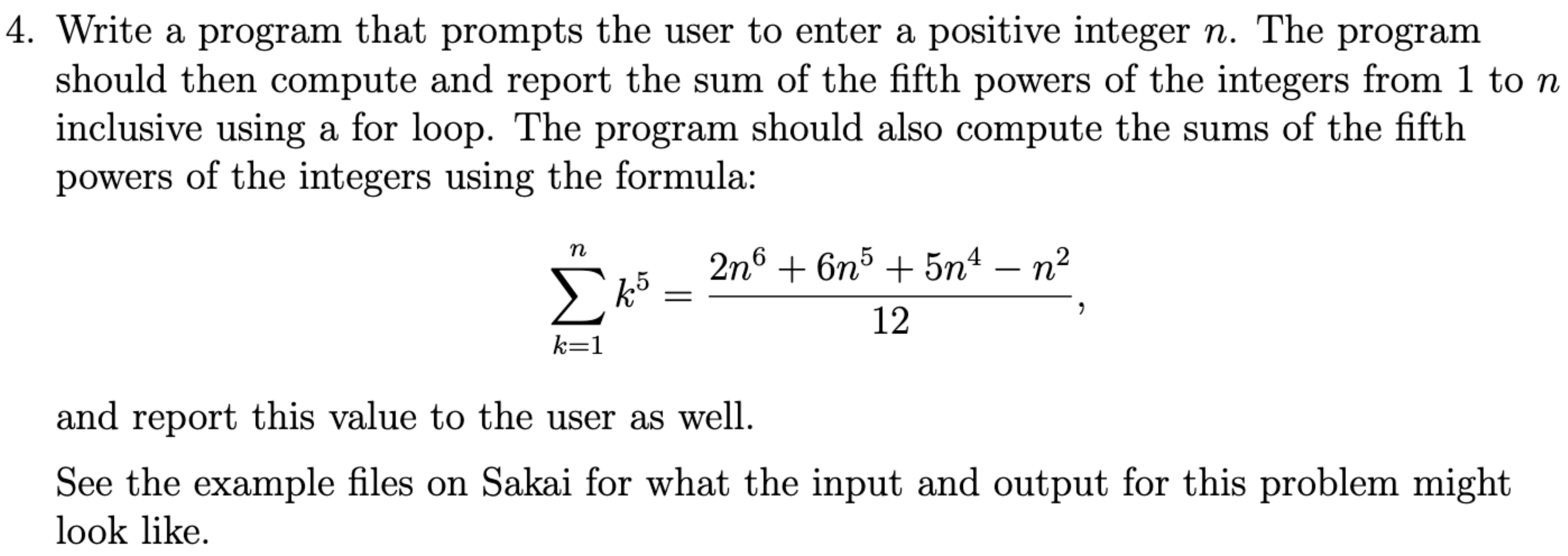
$$|S - S_n| \leq |a_{n+1}|$$

(i.e. the error in approximating the series by the partial sum up to and including the term of index n is less than or equal to the magnitude of the first omitted term).

Let’s make a code

Homework 2

Question 4



4. Write a program that prompts the user to enter a positive integer n. The program should then compute and report the sum of the fifth powers of the integers from 1 to n inclusive using a for loop. The program should also compute the sums of the fifth powers of the integers using the formula:

$$\sum_{k=1}^{n} k^5 = \frac{2n^6 + 6n^5 + 5n^4 - n^2}{12},$$

and report this value to the user as well.

See the example files on Sakai for what the input and output for this problem might look like.

Function

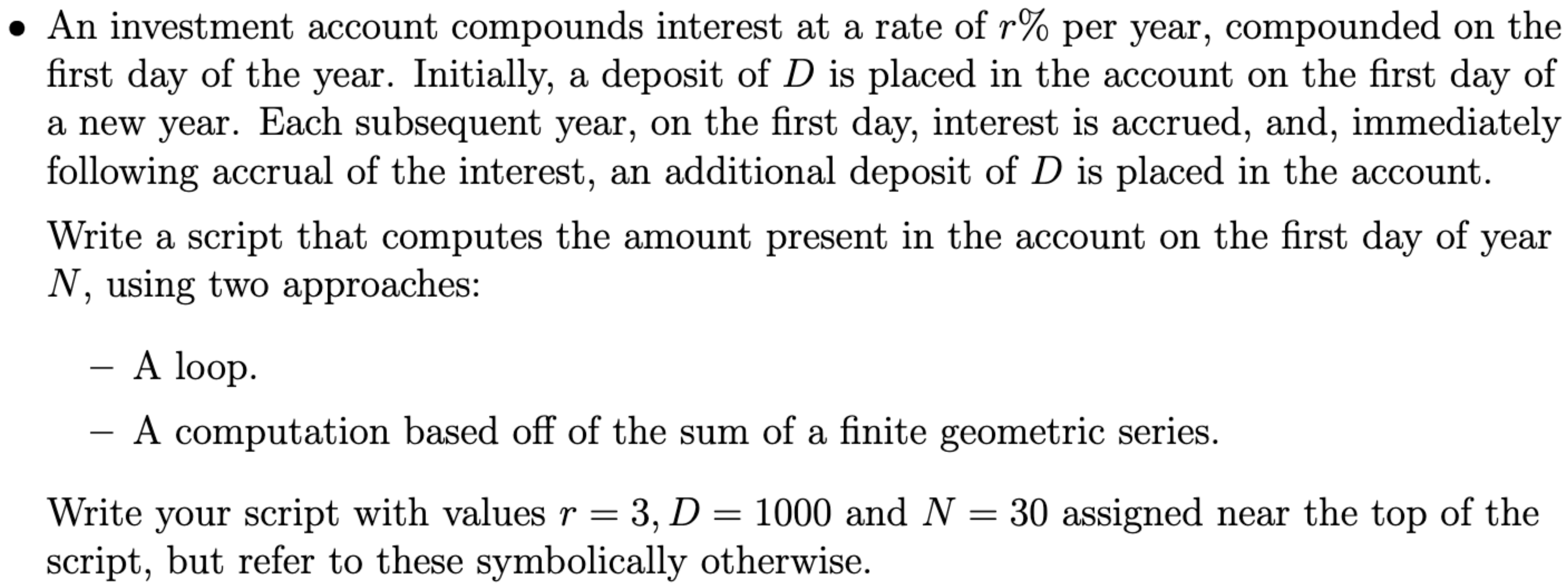
basics

function is for the situation we have to repeat the same operation with different inputs

```
def func_name(input1, input2, …, inputn):
    operations(input1,input2, …, inputn)
    return output

func_name(input_1, input_2, …, input_n)
```

Homework

Extra Question



- An investment account compounds interest at a rate of $r\%$ per year, compounded on the first day of the year. Initially, a deposit of D is placed in the account on the first day of a new year. Each subsequent year, on the first day, interest is accrued, and, immediately following accrual of the interest, an additional deposit of D is placed in the account.

  Write a script that computes the amount present in the account on the first day of year N, using two approaches:

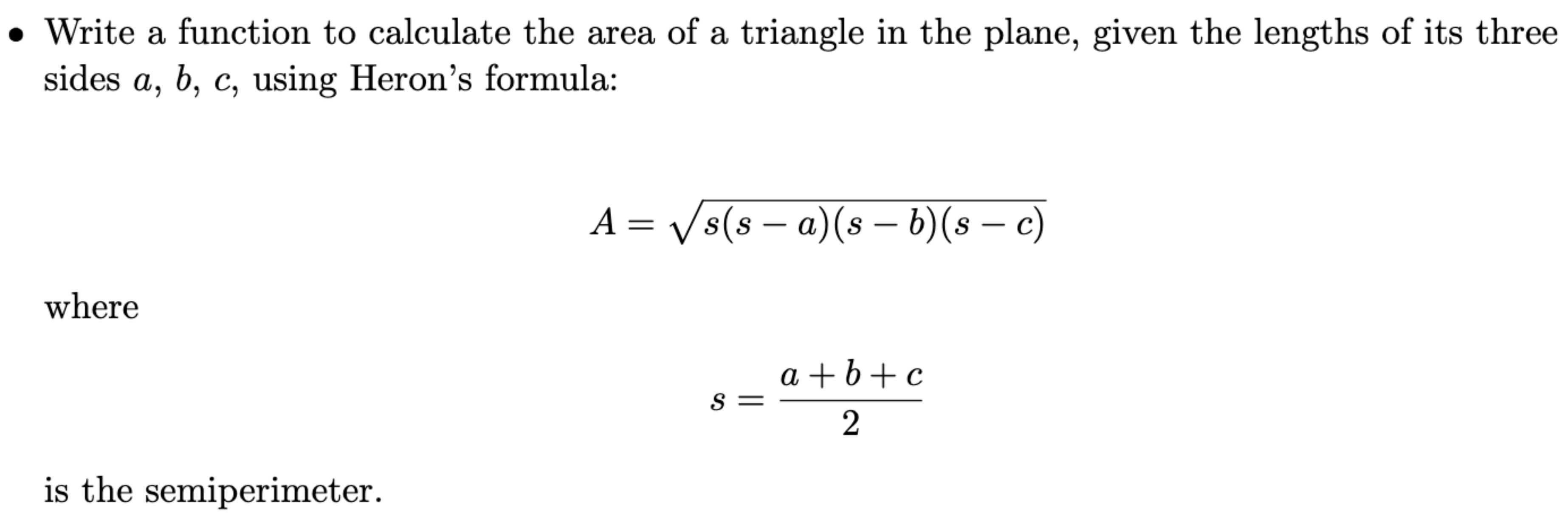
  – A loop.

  – A computation based off of the sum of a finite geometric series.

  Write your script with values $r = 3, D = 1000$ and $N = 30$ assigned near the top of the script, but refer to these symbolically otherwise.

Function

Example



- Write a function to calculate the area of a triangle in the plane, given the lengths of its three sides a, b, c, using Heron's formula:

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

where

$$s = \frac{a+b+c}{2}$$

is the semiperimeter.

1. What should we take as an input?

2. What should be the operation?

3. What should we make it?

Function

basics

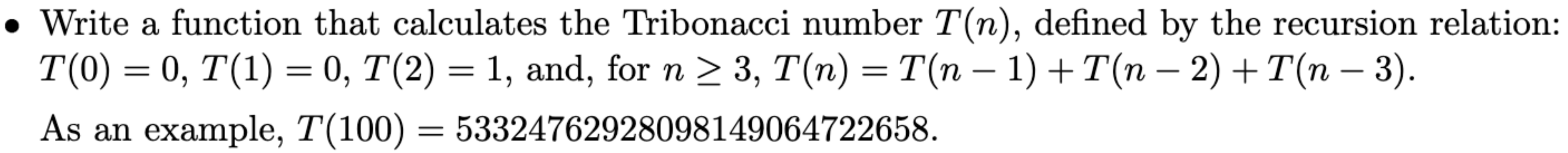
function is for the situation we have to repeat the same operation with different inputs

```
def func_name(input1=default1, input2=default2, input3, …, inputn):

    operations(input1,input2, …, inputn)

    return output
```

```
func_name(input3=?, …, inputn=?)

func_name(input1=?,input2 = ?, input3 = ? …, inputn=?)
```

Function

Example



- Write a function that calculates the Tribonacci number $T(n)$, defined by the recursion relation: $T(0) = 0$, $T(1) = 0$, $T(2) = 1$, and, for $n \geq 3$, $T(n) = T(n-1) + T(n-2) + T(n-3)$.

  As an example, $T(100) = 53324762928098149064722658$.

1. What should we take as an input?

2. What should be the operation?

3. What should we make it?