

# CS 101 Computer Programming and Utilization

## Practice Problems

Param Rathour

<https://paramrathour.github.io/CS-101/Problems.pdf>

Autumn Semester 2020-21

Last update: 2020-12-17 02:01:57+05:30

### Disclaimer

These are **optional** problems. As these problems are pretty involving, my advice to you would be to first solve exercises given in slides and get comfortable with the course content. The taught methods will suffice to solve these problems. (You are free to use 'other' stuff but not recommended)

### Contents

1 Practice Problems 1	2
2 Practice Problems 2	3

## §1. Practice Problems 1

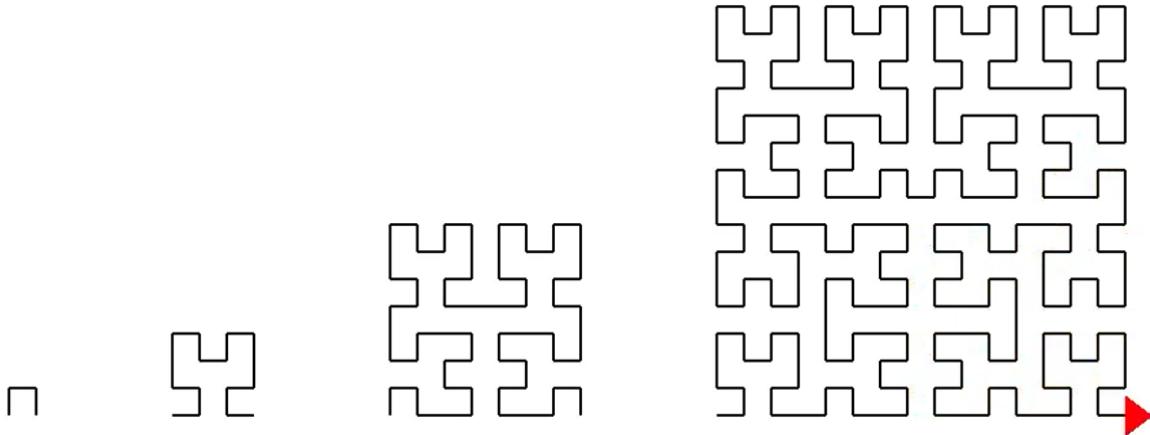


Figure 1: Hilbert Curve

### 1. Problem Statement:

Take an integer as input and draw the corresponding iteration of this fractal using turtleSim  
You may think along these lines

**Step 1** Find a simple pattern in these iterations

**Step 2** Think how can you implement this pattern in an efficient way (here think in the number of lines of code you have to write. **Word of caution:** this is just one of the possible definitions of efficient code)

**Step 3** Do you think that you need something that will implement/shorten your code?  
How will it look like? (it's a feature)

Feel free to discuss your thoughts on this.

**Note.** For people comfortable with the basics of C++, this shouldn't be difficult. You may try this

### Fun Videos

[Hilbert's Curve: Is infinite math useful?](#)

[Recursive PowerPoint Presentations \[Gone Fractal!\]](#)

### Book Chapters for Graphics

Additional chapters of the book on Simplecpp graphics demonstrating its power  
(It is just a list, you are not expected to understand/study things, CS101 is for a reason :P)

**Chapter 5** Coordinate based graphics, shapes besides turtles

**Chapter 15.2.3** Polygons

**Chapter 19** Gravitational simulation

**Chapter 20** Events, Frames, Snake game

**Chapter 24.2** Layout of math formulae

**Chapter 26** Composite class

**Chapter 28** Airport simulation

## §2. Practice Problems 2

1. You probably heard about Fibonacci Numbers!

The Fibonacci numbers are the numbers in the integer sequence: (defined by the recurrence relation)

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \quad n \in \mathbb{Z} \quad (\text{They can be extended to negative numbers}) \end{aligned} \tag{1}$$

For any integer  $n$ , the sequence of Fibonacci numbers  $F_i$  taken modulo  $n$  is periodic.

The Pisano period, denoted  $\pi(n)$ , is the length of the period of this sequence.

For example, the sequence of Fibonacci numbers modulo 3 begins:

0, 1, 1, 2, 0, 2, 2, 1, 0, 1, 1, 2, 0, 2, 2, 1, 0, 1, 1, 2, 0, 2, 2, 1, 0, ... ([A082115](#))

This sequence has period 8, so  $\pi(3) = 8$ . (Basically, the remainder when these numbers are divided by  $n$  is a repeating sequence. You have to find the length of sequence)

### Problem Statement:

- (a) Find Pisano period of  $n$  numbers  $k_1 k_2 \dots k_n$

#### Input Format

$n$

$k_1 k_2 \dots k_n$

#### Output Format

$\pi(k_i)$  (each on a newline)

#### Sample Input

3

3 10 25

#### Sample Output

8

60

100

- (b) For  $n$  numbers  $k_1 k_2 \dots k_n$ , find  $\max(\pi(i))$  for  $i = 1, 2, \dots, k_i$  and corresponding  $i$

If there are 2 (or more) such  $i$ 's, output smallest of them

#### Input Format

$n$

$k_1 k_2 \dots k_n$

#### Output Format

$k \pi(k)$  (each pair on a newline)

Here  $k$  is smallest possible integer satisfying  $\pi(k) = \max(\pi(i))$  for  $i = 1, 2, \dots, k_i$

#### Sample Input

4

20 40 60 100

#### Sample Output

10 60

30 120

50 300

98 336

2.

$$\frac{\pi}{2} = \sum_{k=0}^{\infty} \frac{2^k k!^2}{(2k+1)!} \quad (2)$$

**Problem Statement:**

Calculate  $\pi$  till  $k_i^{\text{th}}$  iteration using Equation (2) for  $n$  different natural numbers  $k_1 \ k_2 \ \dots \ k_n$ .  
Give your answers correct to 10 decimal places **Input Format**

$n$   
 $k_1 \ k_2 \ \dots \ k_n$

**Output Format**

Calculated  $\pi$  for  $k_i$  (each on a newline)

**Sample Input**

3  
10  
20  
30

**Sample Output**

3.1411060216  
3.1415922987  
3.1415926533

3. **Simpson's Rule:** a method for numerical integration

$$\int_a^b f(x) dx \approx \frac{\Delta x}{3} (f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 4f(x_{n-1}) + f(x_n)) \quad (3)$$

**Note.** Simpson's rule can only be applied when an odd number of ordinates is chosen.

**Problem Statement:**

Solve Equation (4) giving the answers correct to 7 decimal places (Use 101 ordinates)

$$\int_{0.5}^1 \frac{\sin \theta}{\theta} d\theta \quad (4)$$

**Correct Answer** = 0.4529756