

Practice Exercise: Starry Starry Night

http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html



Reference: Unit #13 Separate Compilation

Week of release: Week 7

Objectives: This exercise is to show how separate compilation is used in program development

Task statement:

This problem is adopted from National Software Competition 2006 for junior college students. (Copyright: NSC 2006.)

Professor Robert A.J. Matthews of the Applied Mathematics and Computer Science Department at the University of Aston in Birmingham, England, has recently described how the positions of stars across the night sky may be used to deduce a surprisingly accurate value of π . This result followed from the application of certain theorems in number theory.

Here, we don't have the night sky, but we can use the same theoretical basis to form an estimate for π .

Given any pair of positive integers chosen from a large set of random numbers, the probability that these two integers having no common factor other than one is $6/\pi^2$.

For example, using this small set of five numbers {2, 3, 4, 5, 6}, there are 10 pairs that can be formed: (2,3), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6) and (5,6). Six of these 10 pairs - (2,3), (2,5), (3,4), (3,5), (4,5), and (5,6) - have no common factor other than one. Using the ratio of the counts as the probability we have:

$6/\pi^2 = 6/10$. Hence the estimated value of π is 3.1623, correct to four decimal places.

As another example, given this set of 10 numbers {32391, 14604, 3902, 153, 292, 12382, 17421, 18716, 19718, 19895}, there are 24 pairs that have no common factor other than one, among a total of 45 pairs. We have:

$6/\pi^2 = 24/45$. Hence the estimated value of π is 3.3541, correct to four decimal places.

Write a program **estimatePi.c** that reads in a positive integer n representing the size of the list, followed by n unique positive integers (representing the random numbers). Your program then prints out an estimate value for π (using **double** type) accurate to 4 decimal places. The set contains at most 50 unique positive integers.

You will be given two files: **gcd.h** which you need to include in your program, and **gcd.o** which you need to compile together with your program. The **gcd.o** object file contains the GCD function which you are required to use in your program.

The header file **gcd.h** and the object file **gcd.o** for the GCD function are provided. Refer to Unit #13 Separate Compilation slides on how to compile your program **estimatePi.c** with the given **gcd.o**. Sample run #1 below also shows an example.

Do not write your own GCD function. You must use the given **gcd.o** as we want you to understand how separate compilation is done.

Note that the object file **gcd.o** file was generated in sunfire, so it only works in sunfire.

(However, as some students want to try it on their own machine and not on sunfire, they have asked for the source code **gcd.c** so that they can generate **gcd.o** on their own machine. For this reason, we are also providing the **gcd.c** program. But you should not write the GCD function in your own program.)

Sample run #1:

```
$ gcc -Wall -lm estimatePi.c gcd.o -o estimatePi
$ estimatePi
3
7 4 10
Estimated pi = 3.0000
```

Sample run #2:

```
5
2 3 4 5 6
Estimated pi = 3.1623
```

Sample run #3:

```
10
32391 14604 3902 153 292 12382 17421 18716 19718 19895
Estimated pi = 3.3541
```