EE231002 Introduction to Programming
Lab03. Solving a Diophantine Equation

Due: Oct. 30, 2021
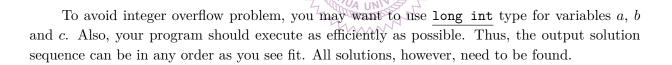
Given a Diophantine equation as

$$a + b^2 = c^3,$$

where a, b, and c are all positive integers. Please write a C program to find all solutions for $1 \le a, b, c \le 5000$.

Example of program execution:

```
$ ./a.out
Sol 1: 2 + 5^2 = 3^3
Sol 2: 4 + 2^2 = 2^3
Sol 3: 7 + 1^2 = 2^3
...
...
Number of solutions found: xxx
```

To avoid integer overflow problem, you may want to use <u>long int</u> type for variables a, b and c. Also, your program should execute as efficiently as possible. Thus, the output solution sequence can be in any order as you see fit. All solutions, however, need to be found.

Notes.

1. Create a directory **lab03** and use it as the working director.

2. Name your program source file as **lab03.c**.

3. The first few lines of your program should be comments as the following

   ```
   // EE231002 Lab03. Solving a Diophantine Equation
   // ID, Name
   // Date
   ```

4. After finishing editing your source file, you can execute the following command to compile it,

```
$ gcc lab03.c
```

If no compilation errors, the executable file, **a.out**, should be generated, and you can execute it by typing

```
$ ./aout
```

5. After you finish verifying your program, you can submit your source code by

```
$ ~ee2310/bin/submit lab03 lab03.c
```

If you see a "submitted" message, the you are done. In case you want to check which file and at what time you submitted your labs, you can type in the following command:

```
$ ~ee2310/bin/subrec lab03
```

It will show the last few submission records.

6. To measure the execution time of your program, the following unix command can be used:

```
$ time ./a.out
```

The **a.out** program is then executed and all the output printed. At the end, an extra line is printed:

```
0.056u 0.004s 0:00.03 66.6%    0+0k 0+0io 0pf+0w
```

The first number is the CPU time it takes to execute **a.out**. It is measured in seconds. The posfix **u** means **user**, not microseconds. An efficient program needs minimum amount of CPU time to execute.