

Euclid's Algorithm

Euclid's Algorithm

The greatest common divisor of two integers, not both zero, is defined to be the largest integer that divides both numbers. This assignment will focus on calculating the greatest common divisor of two integers based on Euclid's Theorem.

Program Specifications

You will write a C++ program to find the greatest common divisor of two numbers m and n based on Euclid's Theorem. Recall that Euclid's Theorem hinges on the fact that $\gcd(m, n) = \gcd(n, m \bmod n)$. Your implementation **must be recursive** and should be able to calculate the greatest common divisor for all inputs for which it is defined; you may assume the input values can be stored in an `int` type. Your program will take in two command line arguments specifying the value of m and n , and will write to standard out the text "`gcd([m],[n]) = [v]`" where `[m]`, `[n]`, and `[v]` are the values of m , n , and $\gcd(m, n)$, respectively. In the case where the greatest common divisor is not defined, you will print the text "`gcd([m],[n]) is undefined`". See the end of this handout for example output.

Submission and Grading

You must use `skeleton1.cpp` (see iLearn) as a starting point for your program, and complete the `calculate_gcd` function; feel free to create any additional helper functions or include any additional standard libraries that you need. Your source code should be contained in a single file and should be named after your TTU email address excluding the "@tnitech.edu" (e.g., `jagraves21.cpp`). All submissions will be made on iLearn — please do not zip or compress your files. Make sure to follow best coding practices (proper naming conventions, useful comments, etc.). Your program should compile without errors or warnings. Programs will be compiled using the following command:

```
g++ -Wall -ansi -pedantic -std=c++11 [source file].
```

Sample Output

The following lines contain sample input and expected output to your programs. Please note that these examples are not exhaustive, and you should verify your programs with additional test cases.

```
$ ./a.out 60 24  
gcd(60,24) = 12
```

```
$ ./a.out -60 -24  
gcd(-60,-24) = 12
```

```
$ ./a.out 7 0  
gcd(7,0) = 7
```

```
$ ./a.out 0 0  
gcd(0,0) is undefined
```