

MSCS 630 – SECURITY ALGORITHMS AND PROTOCOLS

Lab 2



FEBRUARY 3, 2018 THOMPSON RAJAN CWID: 20082947

1. Euclidean Algorithm

Date: 02-03-18

```
* File: Euclidean.java
* Author: Thompson Rajan
* Course: MSCS 630 Security Algorithms and Protocols
* Assignment: Lab 2
* Due Date: Wednesday, February 7, 2018
* Version: 1.0
* This file contains the implementation of the Euclidean Algorithm.
import java.util.Scanner;
* This class implements the Euclidean Algorithm to find the greatest common
* divisor of given two numbers.
public class Euclidean {
  * This method takes two long inputs and return their gcd.
 * @param a - long input
  * @param d - long input
  * @return - returns gcd of a and d, gcd(a,d).
 static long euclidAlg(long a, long d){
  //Get quotient
  long q = Math.floorDiv(a, d);
  //Get remainder
  long r = a - d * q;
  //Call euclidAlg() recursively when remainder is not 0.
  if(r!=0)
   return euclidAlg(d, r);
  return d;
 }
 public static void main(String[] args) {
  Scanner input = new Scanner(System.in);
  //Get input values
  long a = input.nextLong();
  long d = input.nextLong();
  //Get gcd(a,b)
  long r = euclidAlg(a, d);
  System.out.println(r);
```

Thompson Rajan CWID: 20082947

}

Output:

```
Toms-MacBook-Pro:2 tom$ javac Euclidean.java
[Toms-MacBook-Pro:2 tom$ cat input1.a
100 76
[Toms-MacBook-Pro:2 tom$ java Euclidean < input1.a
4
[Toms-MacBook-Pro:2 tom$ cat input2.a
1071 462
[Toms-MacBook-Pro:2 tom$ java Euclidean < input2.a
21
[Toms-MacBook-Pro:2 tom$ cat input3.a
148 75
[Toms-MacBook-Pro:2 tom$ java Euclidean < input3.a
1
Toms-MacBook-Pro:2 tom$ java Euclidean < input3.a
]
Toms-MacBook-Pro:2 tom$ java Euclidean < input3.a
```

Date: 02-03-18 Thompson Rajan CWID: 20082947

2. Extended Euclidean Algorithm:

```
* File: ExtendedEuclidean.java
* Author: Thompson Rajan
* Course: MSCS 630 Security Algorithms and Protocols
* Assignment: Lab 2
* Due Date: Wednesday, February 7, 2018
* Version: 1.0
* This file contains the implementation of the Extended Euclidean Algorithm.
import java.util.Scanner;
* This class implements the Extended Euclidean Algorithm to find the
* co - primes x and y from the equation gcd(a,b) = 1 = ax + by
public class ExtendedEuclidean {
 * This method takes in two long values that are relatively prime, a and b
  * and returns the corresponding the gcd and co - primes x and y.
 * @param a - long input a
  * @param b - long input b
  * @return - long array where u[0] is gcd(a,b), u[1] is 'x' and u[2] is 'y';
 static long[] euclidAlgExt(long a, long b){
  long u[] = \{a, 1, 0\};
  long v[] = \{b, 0, 1\};
  long w[] = new long[v.length];
  long t = 0;
  while(v[0] > 0) {
   //Get floor value of 'a' and 'b'
   t = (long) (Math.floor(u[0] / v[0]));
   //Update w, u and v vectors.
   for(int j = 0; j < v.length; j++)
   {
     w[j] = u[j] - v[j] * t;
     u[j] = v[j];
     v[i] = w[i];
  return u;
 public static void main(String[] args) {
  Scanner input = new Scanner(System.in);
  //Get input values.
  long a = input.nextInt();
  long b = input.nextInt();
  //Get co - primes x and y.
```

Date: 02-03-18

Thompson Rajan CWID: 20082947

```
long u[] = euclidAlgExt(a, b);

System.out.println(u[0] + " " + u[1] + " " + u[2]);
}
}
```

Output:

```
| Toms-MacBook-Pro:2 tom$ javac ExtendedEuclidean.java | Toms-MacBook-Pro:2 tom$ cat input1.b | 148 75 | Toms-MacBook-Pro:2 tom$ java ExtendedEuclidean < input1.b | 1 37 -73 | Toms-MacBook-Pro:2 tom$ cat input2.b | 1155 862 | Toms-MacBook-Pro:2 tom$ java ExtendedEuclidean < input2.b | 1 203 -272 | Toms-MacBook-Pro:2 tom$ cat input3.b | 240 46 | Toms-MacBook-Pro:2 tom$ java ExtendedEuclidean < input3.b | 2 -9 47 | Toms-MacBook-Pro:2 tom$ | Toms-MacBook-Pro:
```

Date: 02-03-18 Thompson Rajan CWID: 20082947