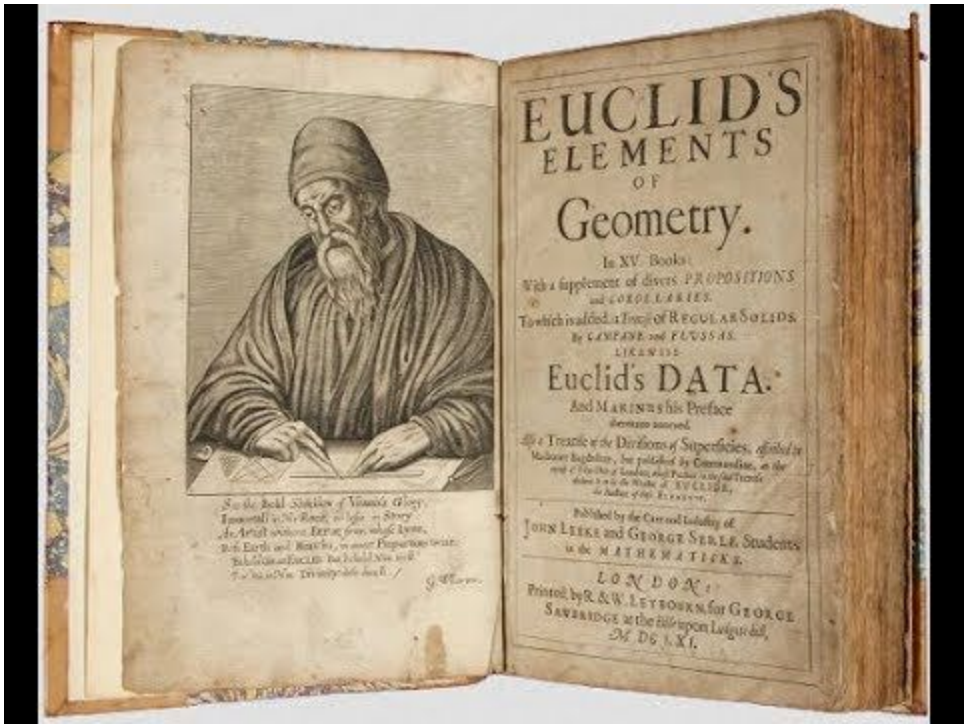
Name: Bernal Leocadio Josué Eduardo.
Teacher: Nidia Asunción Cortez Duarte.
Group: 3CM9.

Cryptography
Euclid's algorithm


EUCLID'S
ELEMENTS
OF
Geometry.
In XV. Books.
With a supplement of divers PROPOSITIONS
and COROLLARIES.
To which is added, a Treatise of REGULAR SOLIDS,
By CAMPANE and FLUSSAS.
LIKEWISE
Euclid's DATA.
And MARINUS his Preface
thereunto annexed.
Also a Treatise of the Divisions of Superficies, applied to
Mechanick Exercises, but published by Commandine, at the
Published by the Care and Industry of
JOHN LEEKE and GEORGE SERLE, Students
in the MATHEMATICKS.
LONDON:
Printed by R. & W. LEYBOURN, for GEORGE
SAWBRIDGE at the Bible upon Ludgate-hill,
M DC LXI.

Introduction.

The Euclidean algorithm, also called Euclid's algorithm, is an algorithm for finding the greatest common divisor of two numbers "a" and "b". The algorithm can also be defined for more general rings than just the integers . There are even principal rings which are not Euclidean but where the equivalent of the Euclidean algorithm can be defined. The algorithm for rational numbers was given in Book VII of Euclid's Elements.

The Euclidean algorithm is an example of a P-problem whose time complexity is bounded by a quadratic function of the length of the input values.

Literature review.

Let $a = bq + r$ then find a number u which divides both a and b (so that $a = su$ and $b = tu$), then u also divides r since.

$$r = a - bq = su - qtu = (s - qt)u.$$

Similarly, find a number v which divides b and r (so that $b = śv$ and $r = tű$), then v divides a since

$$a = bq + r = s'vq + t'v = (s'q + t')v.$$

Therefore, every common divisor of a and b is a common divisor of b and r , so the procedure can be iterated as follows:

$$
\begin{array}{lll}
q_1 = \left\lfloor \frac{a}{b} \right\rfloor & a = b\,q_1 + r_1 & r_1 = a - b\,q_1 \\
q_2 = \left\lfloor \frac{b}{r_1} \right\rfloor & b = q_2\,r_1 + r_2 & r_2 = b - q_2\,r_1 \\
q_3 = \left\lfloor \frac{r_1}{r_2} \right\rfloor & r_1 = q_3\,r_2 + r_3 & r_3 = r_1 - q_3\,r_2 \\
q_4 = \left\lfloor \frac{r_2}{r_3} \right\rfloor & r_2 = q_4\,r_3 + r_4 & r_4 = r_2 - q_4\,r_3 \\
q_n = \left\lfloor \frac{r_{n-2}}{r_{n-1}} \right\rfloor & r_{n-2} = q_n\,r_{n-1} + r_n & r_n = r_{n-2} \\
 & & -q_n\,r_{n-1} \\
q_{n+1} = \left\lfloor \frac{r_{n-1}}{r_n} \right\rfloor & r_{n-1} = q_{n+1}\,r_n + 0 & r_n = r_{n-1}\,/\,q_{n+1}
\end{array}
$$

For integers, the algorithm terminates when q_{n+1} divides r_{n-1} exactly, at which point r_n corresponds to the greatest common divisor of a and b, $\mathrm{GCD}\,(a, b) = r_n$. For real numbers, the algorithm yields either an exact relation or an infinite sequence of approximate relations.

An important consequence of the Euclidean algorithm is finding integers x and y such that

$$a\,x + b\,y = \mathrm{GCD}\,(a, b).$$

This can be done by starting with the equation for r_n, substituting for r_{n-1} from the previous equation, and working upward through the equations.

The extended Euclidean algorithm

Here's a true statement:

If a and b are positive integers, then there are always integers m and n so that the GCD of a and b equals m·a+n·b.

The extended Euclidean algorithm (described, for example, here, allows the computation of multiplicative inverses mod P. First let's see an example. Since the GCD of 210 and 45 is 15, we should be able to write 15 as a sum of multiples of 210 and 45. Here's how to do it. We look carefully at the steps above and change them each.

- Divide 210 by 45, and get the result 4 with remainder 30, so $210=4\cdot45+30$.
- **$30=1\cdot210-4\cdot45$.**
- Divide 45 by 30, and get the result 1 with remainder 15, so $45=1\cdot30+15$.
- **$15=45-1\cdot30=45-1\cdot(1\cdot210-4\cdot45)=-1\cdot210+5\cdot45$**

The greatest common divisor of 210 and 45 is 15, and we have written 15 as a sum of integer multiples of 210 and 45.

The extended Euclidean algorithm has a very important use: finding multiplicative inverses mod P. Choose a prime, P: how about 97. I know 97 is prime, because 2 and 3 and 5 and 7 and even 11 aren't factors of 97, and I only need to check division by primes up to the square root of 97.

Now let me take a fairly random integer, say 20. Since 20 is less than 97, and 97 is prime, the GCD of 20 and 97 should be 1. (Remember, since 97 is prime, its only divisors are itself and 1.) I will verify this by "running" the Euclidean algorithm:

- $97=4\cdot20+17$
- $20=1\cdot17+3$
- $17=5\cdot3+2$
- $3=1\cdot2+1$

The extended Euclidean algorithm allows us to write 1 as a sum of 97 and 20. Here we go:

- **17=1·97-4·20**
- 20-1·17=3 so **3**=1·20-1·17=1·20-**(**1·97-4·20**)**=**-1·97+5·20**
- 17=5·3+2 so **2**=17-5·3=**(**1·97-4·20**)**-5**(**-1·97+5·20**)**=**6·97-29·20**
- **1**=3-2=**(**-1·97+5·20**)**-**(**6·97-29·20**)**=**-7·97+34·20**

The final equation tells us that **1=-7·97+34·20**, which means that the product of 34 and 20 is equal to 1 plus a multiple of 97. But in mod 97, we *ignore* multiples of 97. Therefore 34 is the multiplicative inverse of 20 mod 97.

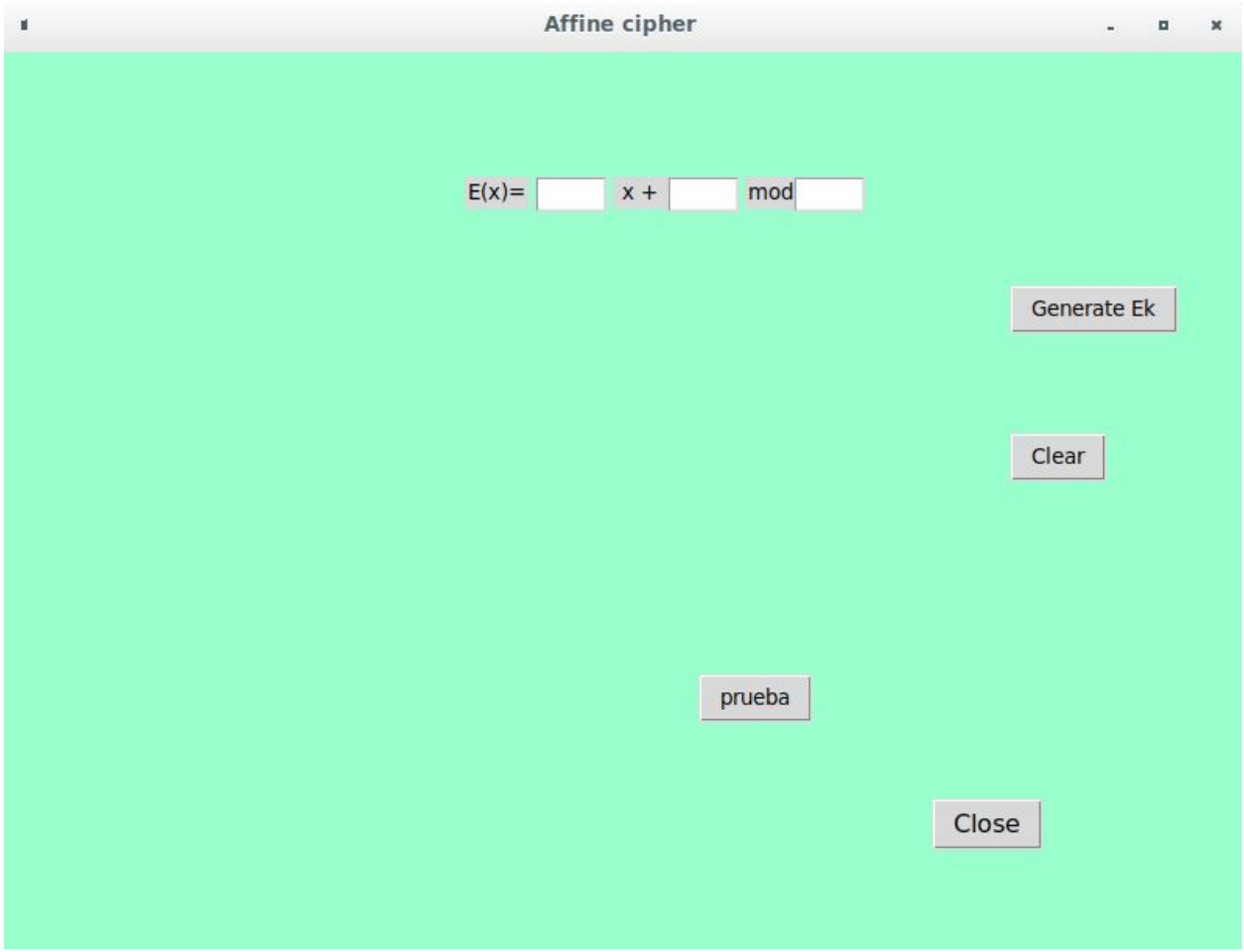
Software.

Python 2.7.15rc1.

Tkinter (GUISś python).

Procedure.

We started a simple graphical user interface, with the following components.

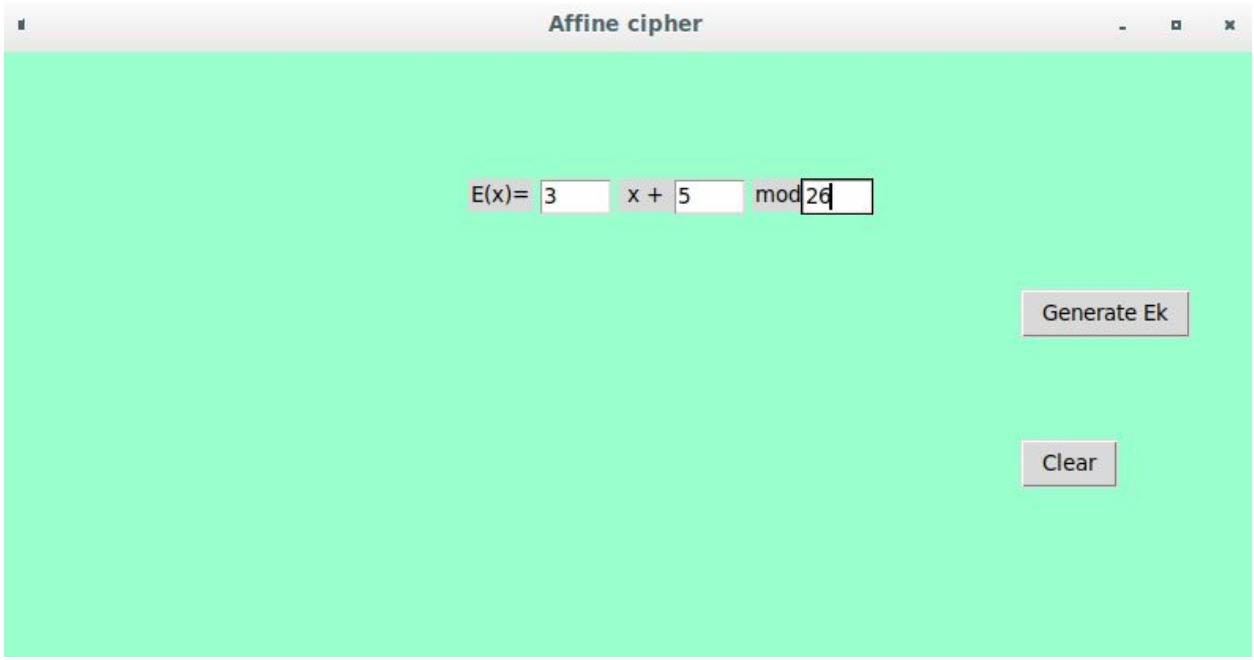


inputs:

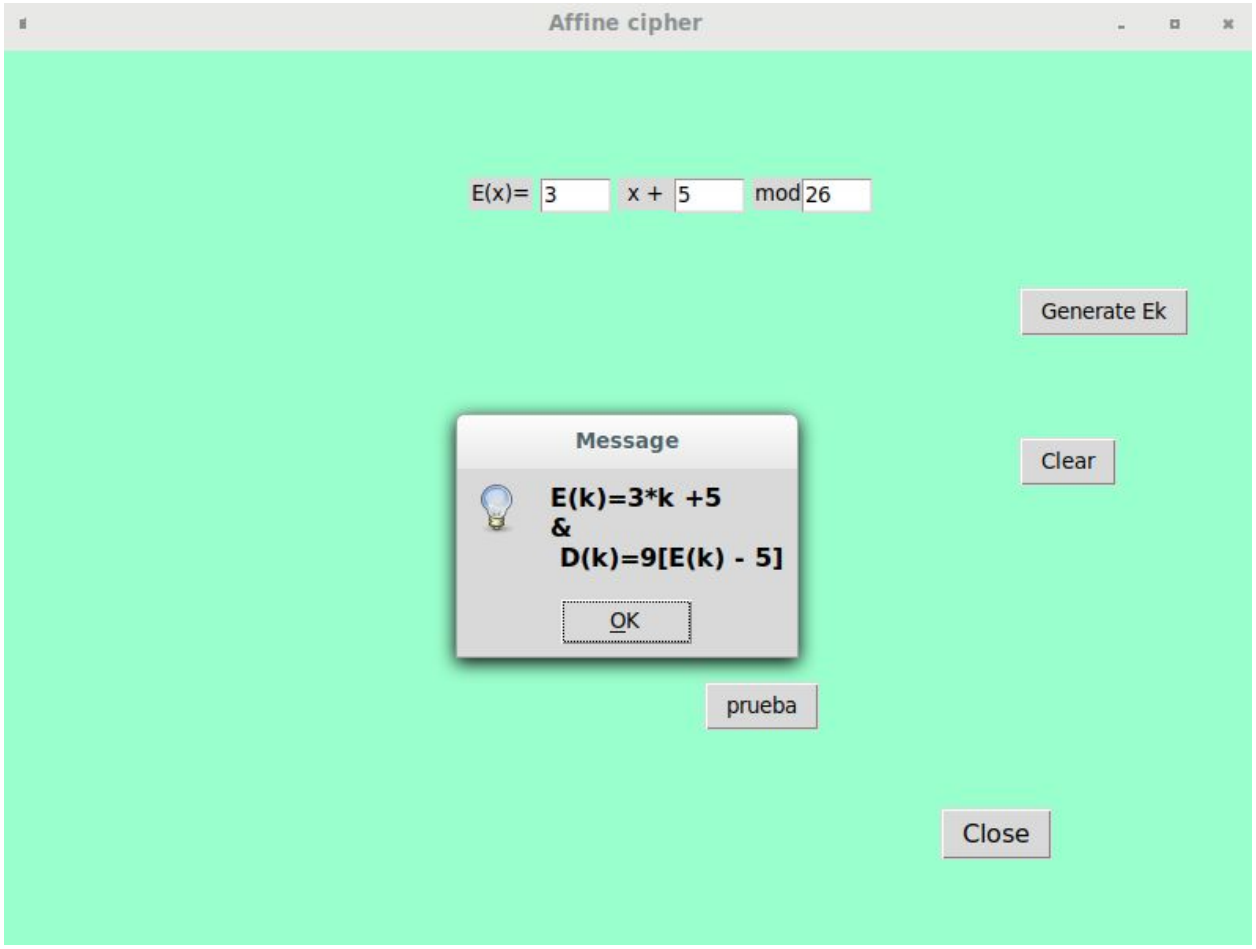
alpha, betha, n. these are the constants of the equation of affine cipher.

output:

Message of alert with the function to decrypt the affine cipher.

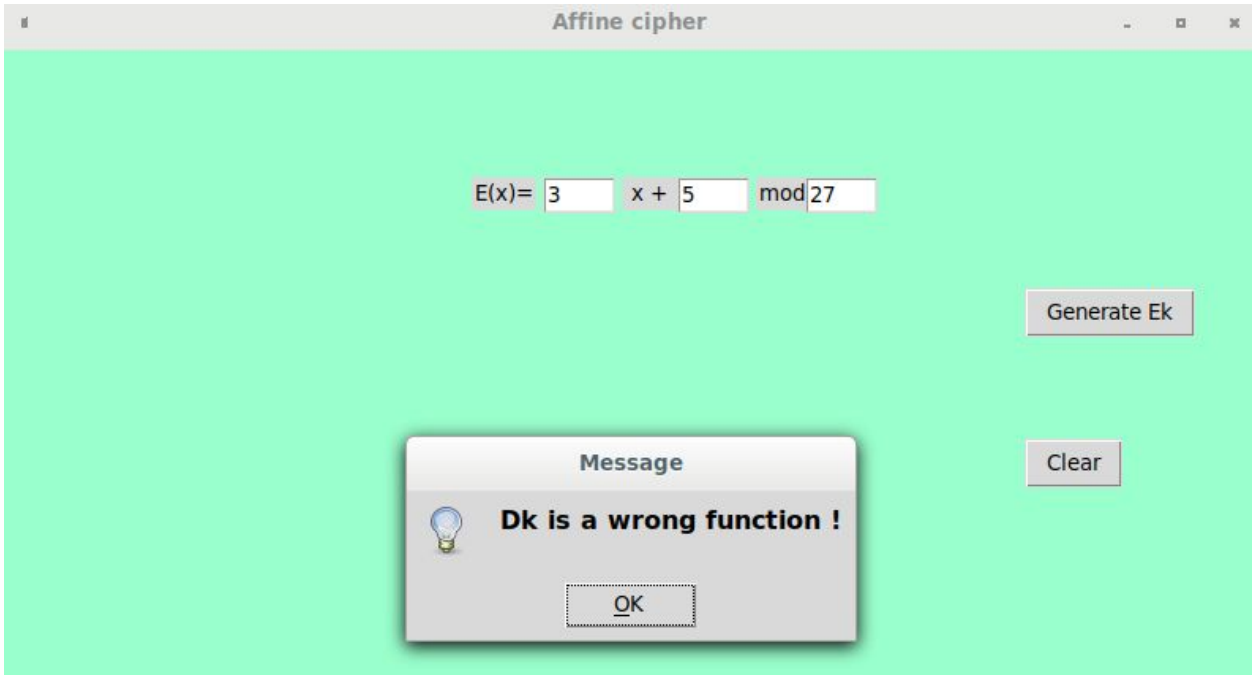


after press the button Generated key.



The condition of exist of the function to decrypt is that $\alpha^{-1} \mod n = 1$, i.e; $\text{MCD}(\alpha, n) = 1$.

If the input values do not satisfy this condition, the application shows the following alert message:



Important functions.

The function return true if MCD(a, b) is 1 in another case return false, with this condition we be sure that exist the Inverse multiplicative.

```
def hasInverse(a,b):

    while a>0:
        t=a
        a=b%a
        b=t
    if b==1:
        return True
    else:
        return False
```

This function return the inverse multiplicative

```
20
21 def inverse(a, p):
22      u=a
23      v=p
24      x1=1
25      x2=0
26      while u!=1:
27           q=v/u
28           r=v-q*u
29           x=x2-q*x1
30           v=u
31           u=r
32           x2=x1
33           x1=x
34      return x1%p
```

Conclusions.

Mathematics are essential for many areas of engineering, particularly in computer science they are used to create algorithms. in this case, the euclid algorithm is used to create decryption functions (affine cipher), later it will be seen that this algorithm is also used by the RSA algorithm.