ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΣΦΑΛΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΔΙΚΤΙΩΝ

ΥΛΟΠΟΙΗΣΗ EL-GAMAL ΚΑΙ ΑΛΓΟΡΙΘΜΟΥ ΚΑΙΣΑΡΑ ΣΕ PYTHON

Περιεχόμενα

.....2
Αλγόριθμος Καίσαρα.....3
Σπάζοντας τον κώδικα.....4
Κώδικας σε Python.....5
Επίθεση με χρήση ωμής βίας.....6
Στιγμιότυπα Εκτέλεσης Καίσαρα.....7
Αλγόριθμος El-Gamal.....9
Παραγωγή κλειδιών.....9
Κρυπτογράφηση.....9
Αποκρυπτογράφηση.....9
Υλοποίηση Αλγορίθμου El-Gamal σε Python.....10
Κώδικας.....10
Οδηγίες Εκτέλεσης.....11
Γραφικό Περιβάλλον.....12
Αρχικό μενού.....12
Παράδειγμα εκτέλεσης Προγράμματος.....12
ΒΙΒΛΙΟΓΡΑΦΙΑ.....14

Αλγόριθμος Καίσαρα

Ο **Κώδικας του Καίσαρα** είναι μία από τις απλούστερες και πιο γνωστές τεχνικές κωδικοποίησης στην κρυπτογραφία.

Είναι **κώδικας αντικατάστασης** στον οποίο κάθε γράμμα του κειμένου **αντικαθίσταται** από κάποιο άλλο γράμμα με σταθερή απόσταση κάθε φορά στο αλφάβητο.

Για παράδειγμα, με **μετατόπιση 3**, το **Α** θα αντικαθιστούν ταν από το **Δ**, το Β από το Ε, και ούτω καθεξής. Η μέθοδος πήρε το όνομά της από τον Ιούλιο Καίσαρα, ο οποίος την χρησιμοποιούσε στην **προσωπική του αλληλογραφία.**

Το βήμα κωδικοποίησης που εκτελείται από τον κώδικα του Καίσαρα συχνά ενσωματώνεται ως τμήμα ενός πιο πολύπλοκου πλαισίου όπως ο κώδικας Vigenère (Βιζενέρ), και έχει ακόμη σύγχρονη εφαρμογή στο σύστημα ROT13.

Όπως με όλους τους μονοαλφαβητικούς κώδικες αντικατάστασης, ο κώδικας του Καίσαρα σπάει εύκολα και στη σύγχρονη εφαρμογή του δεν παρέχει ουσιαστικά κάποια ασφάλεια επικοινωνίας.

Ο μετασχηματισμός μπορεί να αναπαρασταθεί με παράλληλη παράθεση δύο αλφαβήτων. Τα αλφάβητο κωδικοποίησης είναι το απλό αλφάβητο περισταλμένο δεξιά ή αριστερά κατά κάποιο αριθμό θέσεων. Για παράδειγμα ακολουθεί ένας κώδικας του Καίσαρα που χρησιμοποιεί αριστερή περιστροφή τριών θέσεων (η παράμετρος μετατόπισης, εδώ 3, χρησιμοποιείται ως κλειδί):

```
Απλό:     ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
Κώδικας:  ΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩΑΒΓ
```

Η κρυπτογράφηση μπορεί να αναπαρασταθεί με την χρήση αριθμητικής υπολοίπων αν πρώτα μετασχηματιστούν τα γράμματα σε αριθμούς, σύμφωνα με τον κανόνα, Α = 0, Β = 1,.... Ω = 23

(Υπάρχουν διαφορετικοί ορισμοί για την πράξη modulo. Στα παραπάνω το αποτέλεσμα βρίσκεται στο εύρος 0...23. Αν x+n ή x-n δεν βρίσκονται στο εύρος 0...23, αφαιρείται ή προστίθεται 24.)

Η αντικατάσταση παραμένει η ίδια σε όλο το μήνυμα, έτσι ο κώδικας ταξινομείται ως μονοαλφαβητικής αντικατάστασης, σε αντίθεση με τους κώδικες πολυαλφαβητικής αντικατάστασης.

Για το Ceasar Cipher έχουμε

Αποκρυπτογράφηση (Decryption)

- **D(y) = y – k (mod 26)**
- **D(E(x)) = x + k – k (mod 26) = x**

Σπάζοντας τον κώδικα

Ο κώδικας του Καίσαρα μπορεί εύκολα να σπάσει ακόμα και με σενάριο κρυπτοκειμένου μόνο (ciphertext-only scenario).

Μπορούν να ληφθούν υπόψη δύο περιπτώσεις

1.Ο επιτιθέμενος γνωρίζει (ή υποθέτει) ότι έχει χρησιμοποιηθεί κάποιου είδους κώδικας απλής αντικατάστασης, αλλά όχι ότι πρόκειται για τον κώδικα του Καίσαρα συγκεκριμένα.

2.Ο επιτιθέμενος γνωρίζει ότι πρόκειται για κώδικα του Καίσαρα, αλλά δεν γνωρίζει την τιμή της μετατόπισης.

Στην **πρώτη περίπτωση** ο κώδικας μπορεί να σπάσει χρησιμοποιώντας τις ίδιες τεχνικές όπως και σε ένα γενικό απλό κώδικα αντικατάστασης, όπως η **ανάλυση συχνότητας** ή οι **λέξεις μοτίβα** Ενώ θα λύνεται, είναι πιθανό ότι ο επιτιθέμενος θα διαπιστώσει σύντομα την κανονικότητα στη λύση και θα συμπεράνει ότι χρησιμοποιείται ο **κώδικας του Καίσαρα.**

Στη δεύτερη περίπτωση το σπάσιμο του κώδικα είναι ακόμα πιο εύκολο. Καθώς υπάρχει περιορισμένος μόνο αριθμός πιθανών μετακινήσεων (24 στα Ελληνικά), μπορούν να εξεταστούν με τη σειρά **σε μία brute force attack**.Κάτι αντίστοιχο παρουσιάζω στον κώδικα του προγράμματος αλλά με αγγλικό αλφάβητο.

Κώδικας σε Python

Η πρώτη συνάρτηση ειναι η συνάρτηση κρυπτογράφησης στην οποία για κάθε γράμμα του αλφαβήτου και κάνοντας τις ανάλογες μετατροπές σε ASCII επιστρέφω στο κυρίως πρόγραμμα το κρυπτογραφημένο μήνυμα. Για τις ανάγκες της παρουσίασης χρησιμοποίησα το γνωστό απο τις διαφάνειες **“I HAVE A SECRET”**

```
def  caesarEncrypt(text, shiftsNumber):
1. result = ""
2.#Το αποτέλεσμα μου θα καταχωρηθεί στην μεταβλητή result και θα επιστραφεί στο κυρίως πρόγραμμα
3. #Οι μετατοπίσεις ειναι εφικτό να αλλάζουν κάθε φορά
4. #Για λόγους ορθής πιστοποίησης της λειτουργίας του αλγορίθμου χρησιμοποιήσαμε το κλασικό παράδειγμα απο τις διαφάνειες με 3 μετατοπίσεις
5.
6.# Προσπέλασε όλο το κείμενο που σου δίνουμε σαν είσοδο
7. for i in range(len(text)):
8.  #Παίρνει τον κάθε χαρακτήρα στην θέση i
9.  char = text[i]
10.     # Για κάθε κεφαλαίο γράμμα χρησιμοποίησε κρυπτογράφηση
11.
12.  if (char.isupper()):#Εαν είναι κεφαλαίο
13.   result += chr((ord(char) + shiftsNumber - 65) % 26 + 65) #Βρές ποιος χαρακτήρας είναι πρόσθεσε την μετατόπιση και επέστρεψε τον κρυπτο χαρακτήρα
14. # Κρυπτογράφησε τα μικρά γράμματα
15.  else:
16.   result += chr((ord(char) + shiftsNumber - 97) % 26 + 97)
17.
18. return result.replace('q',' ')
```

Με την χρήση 2 συναρτήσεων παίρνω τις ανάλογες εισόδους για το πρόγραμμα και εδώ σημειώνω οτι οι **μετατοπίσεις** που κάνει ο αλγόριθμος ειναι δυναμικές.

```
def  textCipherInput():
1. print(
2.  "Για να τρέξετε το πρόγραμμα με δικό σας κείμενο απλά πληκτρολογήστε, αλλιώς πιέστε ENTER και θα τρέξει παράδειγμα απο Διαφάνειες")
3. text = str(input("Δώστε ένα κείμενο για Κρυπτογράφηση") or "I HAVE A SECRET")
4. return text
```

```
def  shiftsCaesarInput():
1. print(
2.  "Για να τρέξετε το πρόγραμμα με δικό σας αριθμό απο μετατοπίσεις απλά πληκτρολογήστε, αλλιώς πιέστε ENTER και θα τρέξει παράδειγμα απο Διαφάνειες")
3. shiftsNumber = int(input("Δώστε τον αριθμό των Μετατοπίσεων του Αλγορίθμου") or "3")
4. return shiftsNumber
```

Στο τέλος καλώ τη συνάρτηση στο κυρίως πρόγραμμα

```
def  Caesar():
# Έλεγχος και είσοδος Δεδομένων
text=textCipherInput()
shiftsNumber=shiftsCaesarInput()
#text = "I HAVE A SECRET "


print("Αρχικό κείμενο : " + text)
print("Μοτίβο Μετατόπισης : " + str(shiftsNumber))
#Μέτρηση του Αλγορίθμου Χρόνος Εκτέλεσης
from timeit import default_timer as timer
start = timer()
encryptedText=caesarEncrypt(text, shiftsNumber)
end = timer()
print("Κρυπτοκείμενο: ",encryptedText )

print("Χρόνος Εκτέλεσης Κρυπτογράφησης σε nanoseconds  ",round(((end - start)*1000000000)))# Η συνάρτηση επιστρέφει τον χρόνο σε δευτερόλεπτα για λόγους ευκρίνειας το μετατρέπω σε nanoseconds
```

Επίθεση με χρήση ωμής βίας

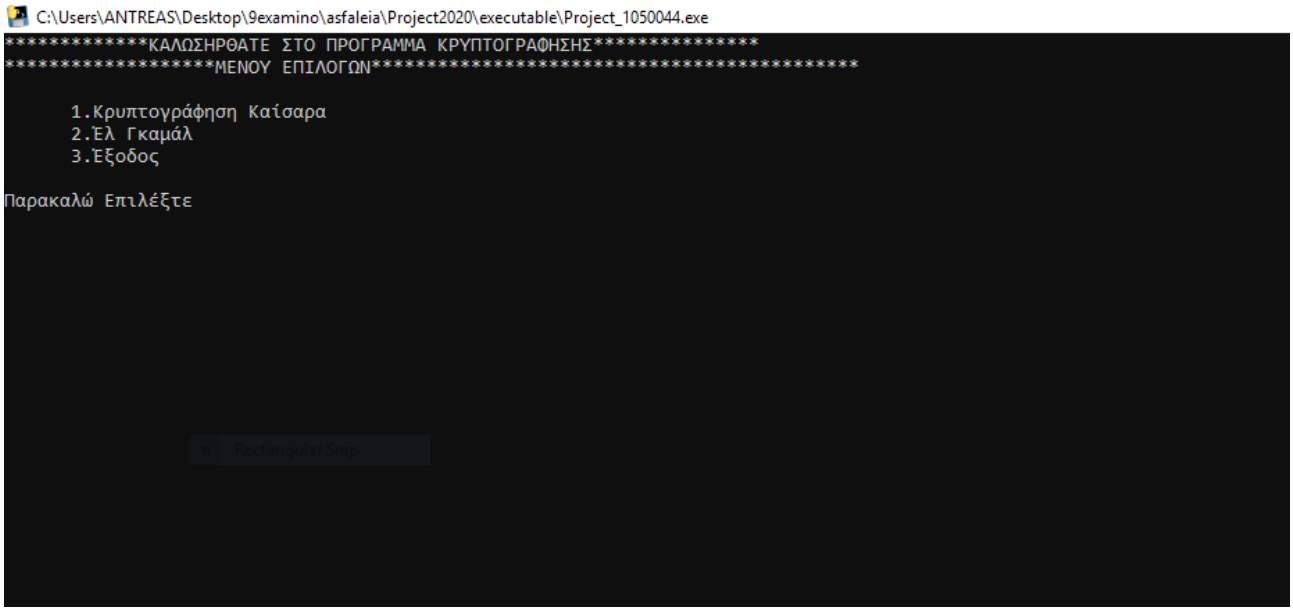
Παρακάτω ειναι ο κώδικας επίθεσης ο οποίος με ωμή βία επιστρέφει μια λίστα απο τα πιθανά κλειδιά στο κυρίως πρόγραμμα.Για την καλύτερη παρουσίαση των αποτελεσμάτων έλαβα και χρόνους αποκρυπτογράφησης με την συνάρηρτηση **timer()**

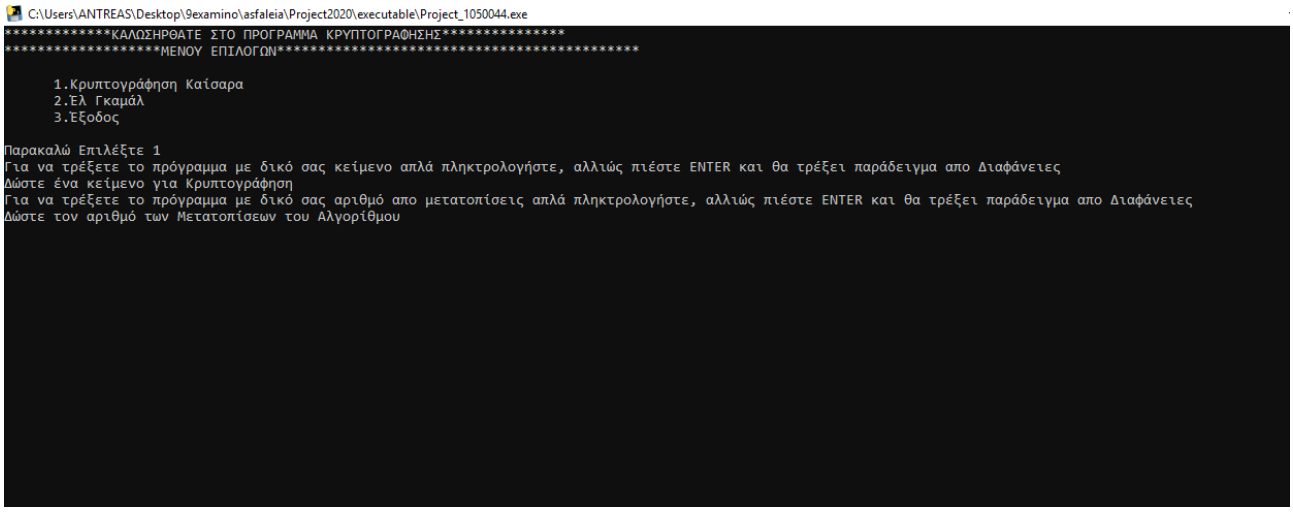
```
letterAlphabet  = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

keylist=[]
descryptionText=''
start = timer()
for key in range(len(letterAlphabet)):

 if (not descryptionText):
  pass
 else:
  keylist.append(descryptionText)
  descryptionText = ''

 for symbol in encryptedText:
  if symbol in letterAlphabet:
   num = letterAlphabet.find(symbol)
   num = num - key
   if num < 0:
    num = num + len(letterAlphabet)
```

```
18.     descryptionText = descryptionText + letterAlphabet[num]
19.   else:
20.     descryptionText = descryptionText + symbol
21. end = timer()
```

Στιγμιότυπα Εκτέλεσης Καίσαρα

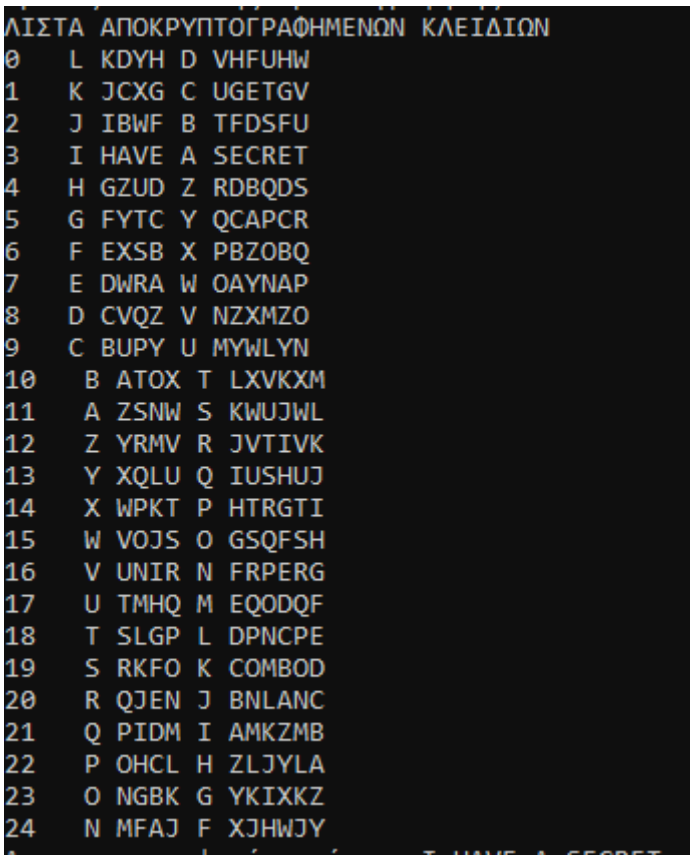
Οπως παρατηρούμε το στο λεξικό περιλαμβάνεται το κωδικοποιημένο κείμενο





C:\Users\ANTREAS\Desktop\9examino\asfaleia\Project2020\executable\Project_1050044.exe

```
Παρακαλώ Επιλέξτε 1
Για να τρέξετε το πρόγραμμα με δικό σας κείμενο απλά πληκτρολογήστε, αλλιώς πιέστε ENTER και θα τρέξει παράδειγμα απο Διαφάνειες
Δώστε ένα κείμενο για Κρυπτογράφηση
Για να τρέξετε το πρόγραμμα με δικό σας αριθμό απο μετατοπίσεις απλά πληκτρολογήστε, αλλιώς πιέστε ENTER και θα τρέξει παράδειγμα απο Διαφάνειες
Δώστε τον αριθμό των Μετατοπίσεων του Αλγορίθμου
Αρχικό κείμενο : I HAVE A SECRET
Μοτίβο Μετατόπισης : 3
Κρυπτοκείμενο:  L KDYH D VHFUHW
Χρόνος Εκτέλεσης Κρυπτογράφησης σε nanoseconds   80298
ΛΙΣΤΑ ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΜΕΝΩΝ ΚΛΕΙΔΙΩΝ
0   L KDYH D VHFUHW
1   K JCXG C UGETGV
2   J IBWF B TFDSFU
3   I HAVE A SECRET
4   H GZUD Z RDBQDS
5   G FYTC Y QCAPCR
6   F EXSB X PBZOBQ
7   E DWRA W OAYNAP
8   D CVQZ V NZXMZO
9   C BUPY U MYWLYN
10   B ATOX T LXVKXM
11   A ZSNW S KWUJWL
12   Z YRMV R JVTIVK
13   Y XQLU Q IUSHUJ
14   X WPKT P HTRGTI
15   W VOJS O GSQFSH
16   V UNIR N FRPERG
17   U TMHQ M EQODQF
18   T SLGP L DPNCPE
19   S RKFO K COMBOD
20   R QJEN J BNLANC
21   Q PIDM I AMKZMB
22   P OHCL H ZLJYLA
23   O NGBK G YKIXKZ
24   N MFAJ F XJHWJY
Αποκρυπτογραφημένο μήνυμα I HAVE A SECRET
Χρόνος Αποκρυπτογράφησης σε nanoseconds   268667

      1.Κρυπτογράφηση Καίσαρα
      2.Ελ Γκαμάλ
      3.Εξοδος

Παρακαλώ Επιλέξτε
```



Αλγόριθμος El-Gamal

Ο αλγόριθμος El-Γamal είναι ένας κρυπτογραφικός αλγόριθμος ασύμμετρης κρυπτογραφίας ο οποίος βασίζεται στη δυσκολία των προβλημάτων διακριτού λογαρίθμου και της ανταλλαγής κλειδιών **Diffie – Hellman**. Η μέθοδος που χρησιμοποιεί είναι με πιθανότητες, έτσι για το ίδιο μήνυμα m, μπορεί να δώσει διαφορετικές κρυπτογραφήσεις. Χρησιμοποιείτε συνήθως για ανταλλαγή κλειδιών συμμετρικής κρυπτογραφίας. Η κρυπτογράφηση με τον αλγόριθμο el-gamal αποτελείτε από **3 βήματα**, την παραγωγή κλειδιού, τη κρυπτογράφηση και την αποκρυπτογράφηση.

Παραγωγή κλειδιών

Για να δημιουργήσει τα κλειδιά του κάθε χρήστη ακολουθεί τα παρακάτω βήματα:

- Διαλέγει πρώτο p και γεννήτορα g του Z^*_p
- Επιλέγουμε έναν αριθμό g ο οποίος πρέπει να είναι generator του
- Διαλέγει τυχαίο d από το {2, ..., p-2}
- Υπολογίζει $g^d \bmod p$
- Δημόσιο κλειδί = {p, g, $g^d \bmod p = h$}
- Ιδιωτικό κλειδί = {x}

Κρυπτογράφηση

Για να στείλει μήνυμα η Αλίκη στη Μπόμπο:

- Βλέπει το κλειδί {p, g, $g^d \bmod p$} του Μπόμπου
- Μετατρέπει το m σε ακέραιο στο διάστημα [1, p-1]
- Διαλέγει τυχαίο k από το {2, ... , p-2}
- Υπολογίζει $c_1 = g^k \bmod p$ και $c_2 = m(g^d)^k \bmod p$
- Στέλνει (c_1, c_2)

Αποκρυπτογράφηση

Για να λάβει ο Μπόμπος το μήνυμα:

- Ο Μπόμπος υπολογίζει $c_1^{p-1-d} \equiv c_1^{-d} \pmod p$
- Υπολογίζει $m \equiv (c_1^{-d}) \cdot c_2 \pmod p$

$$\equiv (g^{-k} \bmod p)^d \cdot m((g^d)^k \bmod p) \pmod p$$

$$\equiv (g^{-k})^d \cdot m(g^d)^k \pmod p$$

Υλοποίηση Αλγορίθμου El-Gamal σε Python

Το πρόγραμμα υλοποιήθηκε στη γλώσσα Python σε περιβάλλον **python3.7** με **pycharm**. Το γραφικό περιβάλλον υλοποιήθηκε στην κονσόλα της Python και με χρήση python-generator δημιούργησα ένα εκτελέσιμο αρχείο το οποίο τρέχει απευθείας σε λογισμικό Windows. Για τις ανάγκες της εκτέλεσης χρησιμοποιήθηκαν και οι βιβλιοθήκες **pow** και **Random** της Python. Για οποιαδήποτε άλλη διανομή λειτουργικού μπορείτε να εκτελέσετε τον κώδικα απο κάποιο ide **εφόσον υποστηρίζει python3 .**

Κώδικας

Στο κατάλογο src μπορείτε να δείτε το αρχείο που αφορά το ενιαίο πρόγραμμα και για τους 2 αλγορίθμους με όνομα **Project_1050044.py**. Όλος ο κώδικας βρίσκετε στο αρχείο αυτό και αποτελείτε από τις εξής συναρτήσεις:

- **def gcd(a,b)**:
  - Συνάρτηση που καλείται στο κυρίως πρόγραμμα για να επιστρέψει τον μέγιστο κοινό διαιρέτη 2 αριθμών

- **def gen_key(q)**:
  - Συνάρτηση που καλείται στο κυρίως πρόγραμμα για να επιστρέψει το κλειδί

- **def power(a, b, c)**:
  - Με χρήση modular αριθμητικής υψώνουμε στη δύναμη για την κρυπτογράφηση
- **def encrypt(msg, q, h, g)**:
  - Συνάρτηση κρυπτογράφησης λαμβάνοντας είσοδο το κείμενο ενα μεγάλο αριθμό τυχαίο το h που καλεί την power και το g
- **def decrypt(en_msg, p, key, q)**:
  - Συνάρτηση αποκρυπτογράφησης με χρήση επαναληπτικού αλγορίθμου με δοκιμές στο κρυπτογραφημένο μήνυμα.
- **def elGamal()**:
  - Αρχή προγράμματος και περιοχή μετρήσεων χρόνων

Για περισσότερες πληροφορίες σχετικά με τις λεπτομέρειες υλοποίησης μπορείτε να δείτε τα σχόλια στο κώδικα.

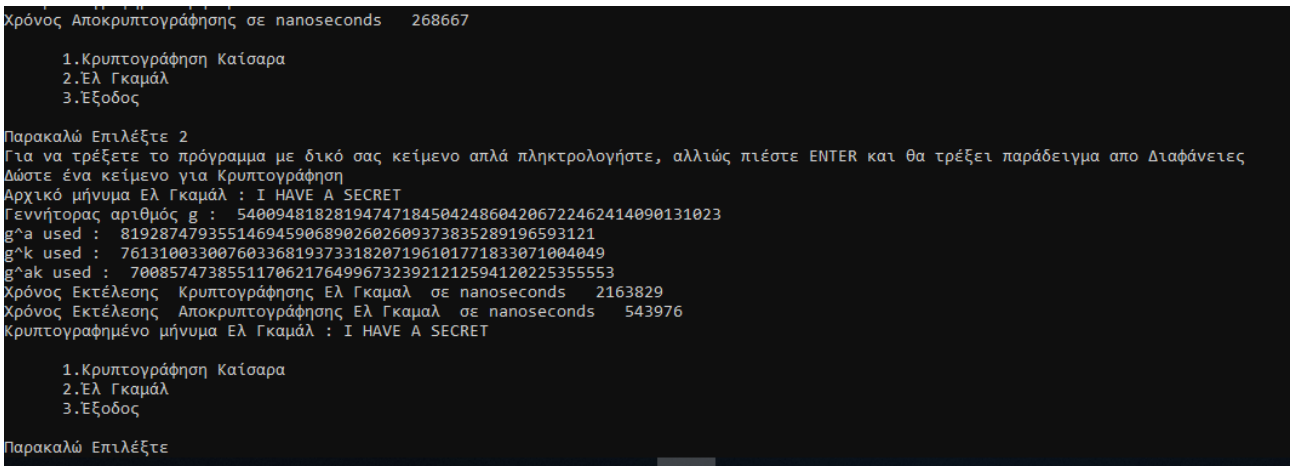
Οδηγίες Εκτέλεσης

Για την εκτέλεση του προγράμματος τρέχουμε το εκτελέσιμο αρχείο Project_1050044 και στη συνέχεια εμφανίζεται το ανάλογο Μενού το οποίο οδηγεί τον χρήστη:

Γραφικό Περιβάλλον

Το πρόγραμμα αποτελείτε από μενού κονσόλας:

Αρχικό μενού

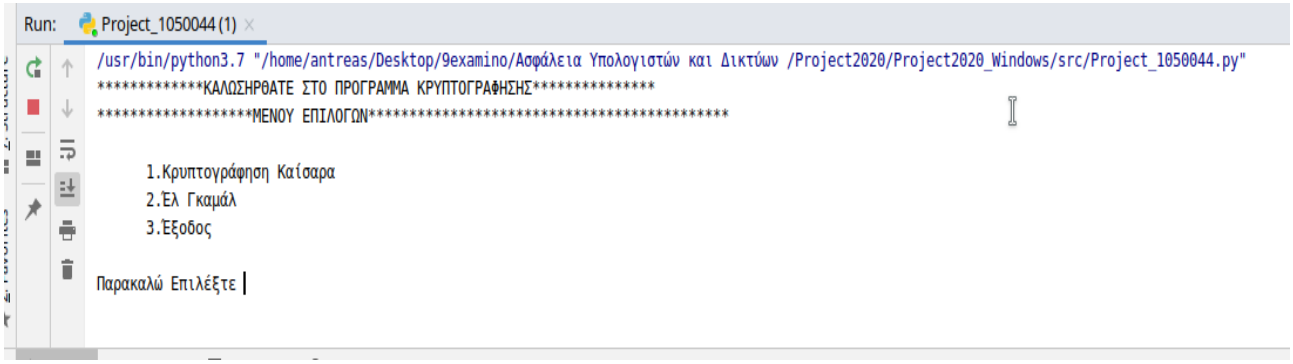


Υπάρχει η επιλογή να τρέξουμε κάποιο δικό μας κείμενο αλλιώς θα τρέξει το κλασσικό παράδειγμα απο τις διαφάνεις “ I HAVE A SECRET”.

Στη συνέχεια βλέπουμε το κρυπτογραφημένο κείμενο μαζί με τα κλειδιά και χρησιμοποιώντας τις βιβλιοθήκες **from timeit import default_timer as timer** λαμβάνω κάποιες μετρήσεις που αφορούν τους **χρόνους εκτέλεσης του Αλγορίθμου**

Παράδειγμα εκτέλεσης Προγράμματος

Έστω ότι ο Μπόμπος θέλει να στείλει στο το μήνυμα **“Hello Alice!!”**. Στιγμιότυπο απο την εκτέλεση του αλγορίθμου.



Για να τρέξετε το πρόγραμμα με δικό σας κείμενο απλά πληκτρολογήστε, αλλιώς πιέστε ENTER και θα τρέξει παράδειγμα απο Διαφάνειες

Δώστε ένα κείμενο για Κρυπτογράφηση *HELLO ALICE!!*

Αρχικό μήνυμα Ελ Γκαμάλ : HELLO ALICE!!

Γεννήτορας αριθμός g : 124289897255624183321373032584710477222929429507

g^a used : 124456375474200522484044895250966352316362665376

g^k used : 167101595844800055333322786638960179135481540804

g^ak used : 188277218762198357599711831355766615805921441820

Χρόνος Εκτέλεσης Κρυπτογράφησης Ελ Γκαμαλ σε nanoseconds 807757

Χρόνος Εκτέλεσης Αποκρυπτογράφησης Ελ Γκαμαλ σε nanoseconds 329290

Κρυπτογραφημένο μήνυμα Ελ Γκαμάλ : HELLO ALICE!!

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Διαφάνειες μαθήματος “ΑΣΦΑΛΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ”-Δημήτριος Σερπάνος-Πανεπιστήμιο Πατρών

[2]”Αλγόριθμος Καίσαρα”- Wikipedia

[3] “ElGamal encryption"-Wikipedia