ΚΡΥΠΤΟΓΡΑΦΙΑ | ΑΛΓΟΡΙΘΜΟΙ ΟΛΙΣΘΗΣΗΣ

ΚΛΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

ΑΛΓΟΡΙΘΜΟΙ ΟΛΙΣΘΗΣΗΣ / ΑΛΓΟΡΙΘΜΟΣ Affine

Εισαγωγή: Με βάση τις προηγούμενες διαλέξεις, αξιοποιήσαμε συναρτήσεις όπως το ord() για την κωδικοποίηση των χαρακτήρων της αλφαβήτου σε αριθμούς και δημιουργήσαμε συναρτήσεις. Στην διάλεξη/εργαστήριο αυτό θα δούμε περισσότερα παραδείγματα συμπεριλαμβάνοντας μια άλλη υλοποίηση του αλγορίθμου ολίσθησης shift, που βασίζεται στη λογική της κρυπτογράφησης του Καίσαρα.

Troubleshooting ideas: Στα Windows 10 η προκαθορισμένη θέση της PYTHON βρίσκεται σε διαφορετικό σημείο από εκεί που την εγκαθιστά το .exe από την ιστοσελίδα. Επίσης υπάρχει η περίπτωση να έχουμε πολλαπλές εκδόσεις PYTHON όπως 2 και 3. Για να επικαλεστούμε την PYTHON v.3 συχνά γράφουμε ως εντολή python3 –arguments. Το ίδιο και με το package installer (pip), μπορεί να μας ζητηθεί να το αναβαθμίσουμε εκτελώντας pip install –ugprade pip.

Functions: Σε προηγούμενα παραδείγματα ορίσαμε συνάρτηση με την γραμμή def Caesar(message). Κατά την εκτέλεση του .py αρχείου η συνάρτηση αυτή θα κάνει load στην Python (μνήμη) και θα μπορούμε πλέον να την εκτελέσουμε. Μπορούμε επίσης στο ίδιο αρχείο να την καλέσουμε για να δούμε το αποτέλεσμα (load function + execute). Ανάλογα την περίπτωση πράττουμε αντίστοιχα.

Κώδικας Python (shift):

```
# Filename: shift.py
def sencode(message):
    message=message.lower()
    msg = list(message)
    for i in range(0,255):
        if i<97 or i>122:
            while chr(i) in msg:
                msg.remove(chr(i))
    z26 = [ord(i)-97 for i in msg]
    return z26
```

Ord(): Η συνάρτηση ord() επιστρέφει έναν ακέραιο που αντιπροσωπεύει τον χαρακτήρα Unicode. (e.g., Το P είναι 80).

Διαχείριση Αρχείων: Στα arguments των συναρτήσεων υπάρχει η ιδιότητα file_selected=False. Με αυτόν τον τρόπο εκτελείται ο αλγόριθμος με 2 διαφορετικές περιπτώσεις. Μπορούμε να τον εκτελέσουμε ως file_selected=True που σημαίνει ότι το cipher text θα αποθηκευτεί σε αρχείο. Το ίδιο μπορεί να γίνει και στην αποκρυπτογράφηση. Τα αρχεία μπορούμε να τα ανοίξουμε πχ. file=open(filename,'a'), και μπορούμε να κάνουμε εγγραφή με την συνάρτηση open('file','w+').

```
# Συνέχεια Filename: shift.py
def sencrypt(p, key, file_selected=False):
    if file_selected==True:
        filename=p
        file=open(filename,'a')
        file.write(p)
        file.close()

    m=sencode(p)
    enc = [(i+key) % 26 for i in m]
    cy = [chr(i+97) for i in enc]
    ctext=''.join(cy)
    if file_selected==False:
        print("\n Αρχικό μήνυμα: %s %r" %(p,m))
        print("\n Κρυπτογράφημα: %s %r" %(ctext,enc))
        return ctext
    else:
        print('Encrypting to file \"ctext.txt\".....\n')
        file=open('ctext.txt','w+')
        file.write(ctext)
        file.seek(0)
        print(file.read())
        file.close()
```

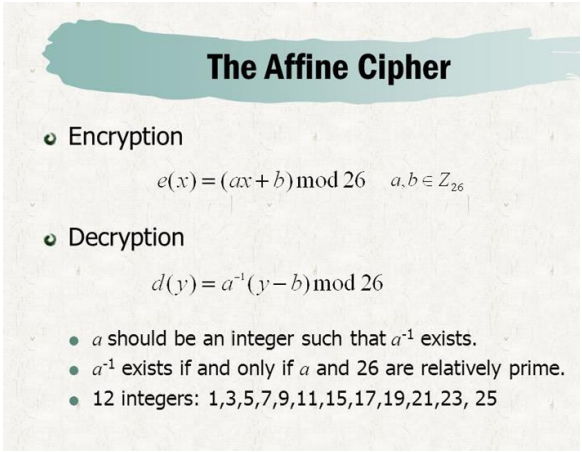
```
# Συνέχεια Filename: shift.py
def sdecrypt(c, key, file_selected=False):
    if file_selected==True:
        filename=c
        file=open(filename,'r')
        c=file.read()
        file.close()
    d=sencode(c)
    dec = [(i - key) % 26 for i in d]
    plain = [chr(i+97) for i in dec]
    ptext=''.join(plain)
    if file_selected==False:
        print("\n Κρυπτογράφημα: %s %r" %(c,d))
        print("\n Αρχικό Μήνυμα: %s %r" %(ptext,dec))
        return ptext
    else:
        print('Decrypting to file  \"ptext.txt\".....\n')
        file=open('ptext.txt', 'w+')
        file.write(ptext)
        file.seek(0)
        print(file.read())
        file.close()
```

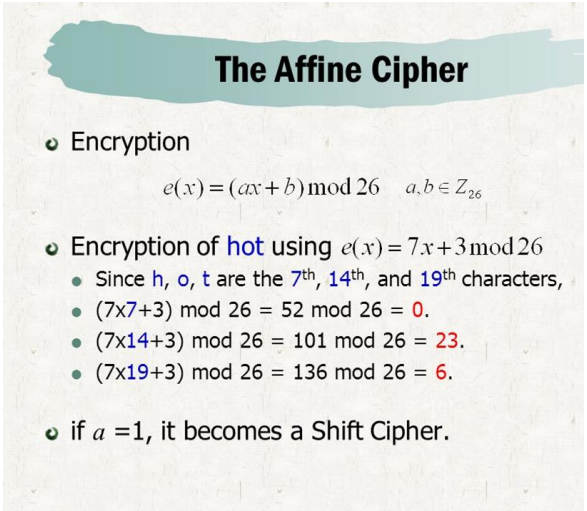
ΠΑΡΑΔΟΤΕΟ 06 (shift.py)

Εκτελέστε τις εντολές δημιουργώντας ένα αρχείο .py και **σχολιάστε σε κάθε γραμμή τι συντελείται. Δώστε δικές σας τιμές για κλειδί και plain text. Εκτελέστε και την αντίστροφη διαδικασία (sdecrypt) με βάση το αποτέλεσμα σπο την sencrypt.**

Ανεβάστε το αρχείο py. Κάντε attach σε zip screenshots από την εκτέλεση των συναρτήσεων (sencode, sencrypt, sdecrypt). Διορθώστε τμήματα κώδικα για την δημιουργία αρχείου αν δεν υπάρχει.

Συμπεριλάβετε screenshots (snipping tool) από την εκτέλεση του κώδικα (4-5 screenshots που να δείχνει το αρχείο που δημιουργήθηκε είτε ως jpg ή σε docx).






```
# Συνέχεια Filename: affine.py
def aencode(message):
    message=message.lower()
    msg = list(message)
    for i in range(0,255):
        if i<97 or i>122:
            while chr(i) in msg:
                msg.remove(chr(i))
    z26 = [ord(i)-97 for i in msg]
    msg=''.join(msg)
    return msg,z26
```

```
# Συνέχεια Filename: affine.py
def gcd(x, y):
    if y==0:
        return x
    else:
        return gcd(y, x % y)

def inv(a):
    if gcd(26, a) == 1:
        for i in range(26):
            if (a * i % 26) == 1:
                return i
    else:
        print("Lathos kleidi")
```

```
# Συνέχεια Filename: affine.py
def aencrypt(msg,a,b,file_selected=False):
    if file_selected==True:
        filename=msg
        file=open(filename,'r')
        msg=file.read()
        file.close()
    plain,z26 = aencode(msg)
    enc = [(a*i + b) % 26 for i in z26]
    cy = [chr(i+65) for i in enc]
    ctext=''.join(cy)
    print("\nH kryptografisi me kleidi (%r, %r) einai:\n
\n %r \n %r" %(a, b, plain, ctext))
    if file_selected==False:
        return ctext
    else:
        file1=open('ctext.txt', 'w')
        file1.write(ctext)
        file1.close()
```

```
# Συνέχεια Filename: affine.py
def adecrypt(msg,a,b,file_selected=False):
    if file_selected==True:
        filename=msg
        file=open(filename,'r')
        msg=file.read()
        file.close()
    ciphertext,z26 = aencode(msg)
    ciphertext=ciphertext.upper()
    a_inv = inv(a)
    dec = [(a_inv * (i - b)) % 26 for i in z26]
    plain = [chr(i+97) for i in dec]
    ptext=''.join(plain)
    print("\nH apokryptografisi me kleidi (%r, %r)
einai:\n \n %r \n %r" %(a,b, ciphertext, ptext))
    if file_selected==False:
        return ptext
    else:
       file1=open('ptext.txt', 'w')
       file1.write(ptext)
       file1.close()
```