11

Κωδικοποίηση ASCII

Κάθε χαρακτήρας (γράμμα, αριθμός, σύμβολο) αντιστοιχίζεται σε έναν μοναδικό κωδικό ASCII (American Standard Code for Information Interchange) σύμφωνα με τον ακόλουθο πίνακα.

32	(space)	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o		

Οι συναρτήσεις chr() και ord()

Ανοίξτε το IDLE και δοκιμάστε τα εξής ενώ ταυτόχρονα παρατηρείτε και τον πιο πάνω πίνακα ASCII.

```
>>> chr(65)
>>> ord('A')
>>> chr(65+8)
>>> chr(52)
>>> chr(ord('F'))
>>> ord(chr(68))
```

Ποιο είναι το συμπέρασμα για την λειτουργία των συναρτήσεων chr() και ord();

11

Κρυπτογράφηση – ένας απλός τρόπος

Κρυπτογράφηση είναι η αλλαγή της μορφής ενός μηνύματος, ώστε κάποιος τρίτος να μην μπορεί να το κατανοήσει.

Θέλουμε ένα φτιάξουμε ένα πρόγραμμα το οποίο θα ζητάει από τον χρήστη τρεις πληροφορίες: (α) mode λειτουργίας, (β) το κλείδι, (γ) το μήνυμα.

```
Press (1) to encrypt a message, (2) to decrypt a message, (3) to exit:  1
Message: Καλημέρα κόσμε!
Key number: 4
Encrypted text:   Ξεολπαυε$ξθχπι%
```

```
Press (1) to encrypt a message, (2) to decrypt a message, (3) to exit:  2
Message: Ξεολπαυε$ξθχπι%
Key number: 4
Decrypted text:   Καλημέρα κόσμε!
```

```
Press (1) to encrypt a message, (2) to decrypt a message, (3) to exit:  2
Message: Ξεολπαυε$ξθχπι%
Key number: 8
Decrypted text:   ΖέηγθΩνέ◷ζψοθα∝
```

```
Press (1) to encrypt a message, (2) to decrypt a message, (3) to exit:  3
```

Αν δεν χρησιμοποιηθεί το ίδιο κλειδί, η αποκρυπτογράφηση δεν θα δώσει το αρχικό μήνυμα.

Η μέθοδος που θέλουμε να εφαρμόσουμε είναι η μετακίνηση των συμβόλων κατά τόσες θέσεις μέσα στο αλφάβητο όσες ορίζει το κλειδί.

Μετακίνηση μέσα στον πίνακα ASCII

Ας υποθέσουμε ότι το κλειδί κρυπτογράφησης είναι το 3. Αυτό σημαίνει ότι το 'α' θα γίνει 'δ', το 'κ' θα γίνει 'ν', κλπ. Κατά πόσο διαφέρουν οι ASCII κωδικοί του 'α' και του 'δ'; Θα πρέπει να διαφέρουν κατά 3. Επιβεβαιώστε το στο IDLE.

```
>>> ord('α')
```

```
>>> ord('δ')
```

Ας δοκιμάσουμε τώρα μερικά μαθηματικά με τους ASCII κωδικούς:

```
>>> key = 3
```

```
>>> symbol = 'α'
```

```
>>> newcode = ord(symbol) + key
```

```
>>> newsymbol = chr(newcode)
>>> newsymbol
```

Τι περιέχει η μεταβλητή newsymbol και γιατί;

Ας δοκιμάσουμε να κρυπτογραφήσουμε μια λέξη. Αυτό σημαίνει ότι πρέπει να πάρουμε κάθε γράμμα της λέξης, να μετατρέψουμε το γράμμα σε ASCII κωδικό, να προσθέσουμε το κλειδί, να μετατρέψουμε τον νέο κωδικό σε γράμμα.

Πως μπορούμε να πάρουμε ένα-ένα τα γράμματα μιας συμβολοσειράς (λέξης) ; Θυμηθείτε την εντολή for.

```
>>> key = 3
>>> word = 'Καλημέρα'
>>> for letter in word:
        code = ord(letter)
        newcode = code + key
        newletter = chr(newcode)
        print(newletter, end='')
```

Τι εμφανίζει η εντολή print();

Ας δοκιμάσουμε να αποκρυπτογραφήσουμε μια λέξη. Αυτό σημαίνει ότι πρέπει να πάρουμε κάθε γράμμα της λέξης, να μετατρέψουμε το γράμμα σε ASCII κωδικό, να αφαιρέσουμε το κλειδί, να μετατρέψουμε τον νέο κωδικό σε γράμμα.

```
>>> key = 3
>>> crypto = 'Νδξκούτδ'
>>> for letter in crypto:
        code = ord(letter)
        newcode = code - key
        newletter = chr(newcode)
        print(newletter, end='')
```

Ο διπλανός κώδικας κρυπτογραφεί.

Το key είναι μια μεταβλητή για το κλειδί.

Το word είναι μια συμβολοσειρά για την αρχική λέξη.

Τι είναι το crypto;

```
>>> key = 3
>>> word = 'Καλημέρα'
>>> crypto = ''
>>> for letter in word:
        code = ord(letter)
        newcode = code + key
        newletter = chr(newcode)
        crypto = crypto + newletter


>>> crypto
'Νδξκούτδ'
```

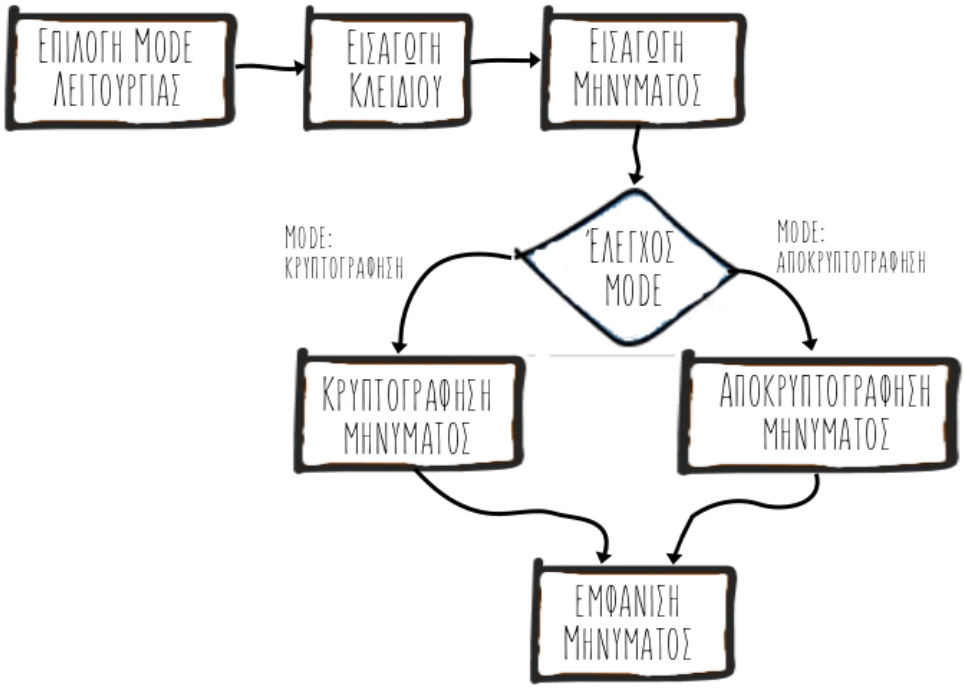
11

Το πρόγραμμα

Προδιαγραφές - Τι θέλουμε να κάνει το πρόγραμμα: Ο χρήστης θα επιλέγει mode λειτουργίας (κρυπτογράφηση, αποκρυπτογράφηση). Θα δίνει το κλειδί και το μήνυμα (είτε το κανονικό είτε το κρυπτογραφημένο, αναλόγως του τι θέλει να κάνει). Το πρόγραμμα θα μετατρέπει το μήνυμα στην ζητούμενη μορφή (προσθέτοντας ή αφαιρώντας το κλειδί από τον ASCII κωδικό κάθε συμβόλου).

Σχεδίαση του προγράμματος – Τμηματικός Προγραμματισμός: Πρέπει να σκεφθούμε ποιες είναι οι αυτόνομες μικρές εργασίες στις οποίες μπορεί να διασπαστεί η συνολική λογική του προγράμματος.

Για παράδειγμα: η επιλογή σωστού mode λειτουργίας, η επιλογή σωστού κλειδιού, η κρυπτογράφηση ενός μηνύματος, η αποκρυπτογράφηση ενός μηνύματος.



Κάθε μια από αυτές τις μικρές εργασίες θα γίνει συνάρτηση (function).

Κώδικας – Βήμα 1ο: Επιλογή σωστού mode λειτουργίας

Οι αποδεκτές τιμές είναι 1 για κρυπτογράφηση, 2 για αποκρυπτογράφηση, 3 για έξοδο. Η function θα επιστρέφει την επιλογή του χρήστη (με την εντολή return).

Σε ένα νέο αρχείο, γράφουμε τον κώδικα της συνάρτησης καθώς και ένα δοκιμαστικό κυρίως πρόγραμμα, για να ελέγξουμε ότι η συνάρτηση επιστρέφει σωστά την επιλογή του χρήστη.

Δείτε τον πιο κάτω ημιτελή κώδικα.

11

crypto1.py

File Edit Format Run Options Window Help

```
def getMode():
    mode = .....    # μία αρχική άσχετη τιμή
    while mode < 1 ..... mode > 3:    # and ή or
        print('Επιλέξτε:')
        print('1 για κρυπτογράφηση')
        print('2 για αποκρυπτογράφηση')
        print('3 για έξοδο')
        mode = .....  # να δίνει τιμή ο χρήστης
        mode = int(mode)

    return .....


# κυρίως πρόγραμμα για δοκιμή
a = getMode()
print('Επιλογή: ', a)
```

Κώδικας – Βήμα 2ο: Εισαγωγή Κλειδιού

Σκεφθείτε κάποιον περιορισμό για την τιμή του κλειδιού: πχ να είναι μεταξύ 1 και 10, να είναι το πολύ όσα τα γράμματα του αλφάβητου, κλπ.

Δημιουργείστε τη συνάρτηση getKey(), η οποία θα δουλεύει περίπου όπως η getMode(), και θα επιστρέφει το κλειδί που επέλεξε ο χρήστης.

Στο δοκιμαστικό κυρίως πρόγραμμα, καλέστε την getKey() για να ελέγξετε ότι επιστρέφει σωστά το κλειδί.

Κώδικας – Βήμα 3ο: Κρυπτογράφηση Μηνύματος

Η συνάρτηση αυτή κρυπτογραφεί ένα μήνυμα. Άρα χρειάζεται ως είσοδο (παράμετροι μέσα στην παρένθεση) δύο πράγματα: το μήνυμα, το κλειδί.

Αυτό που επιστρέφει (εντολή return) είναι το κρυπτογραφημένο μήνυμα.

Τι είναι το crypto;

Τι ακριβώς κάνει η εντολή

crypto = crypto + newsymbol

Τι θα συμβεί αν σβήσουμε την εντολή

crypto = ‘’

```
def encrypt(message, key):
    crypto = ''
    for letter in message:
        newletter = ord(letter) + key
        newsymbol = chr(newletter)
        crypto = crypto + newsymbol

    return(crypto)
```

Κώδικας – Βήμα 4ο: Αποκρυπτογράφηση Μηνύματος

Δημιουργείστε τη συνάρτηση decrypt(crypto, key), η οποία είναι πολύ παρόμοια με την encrypt().

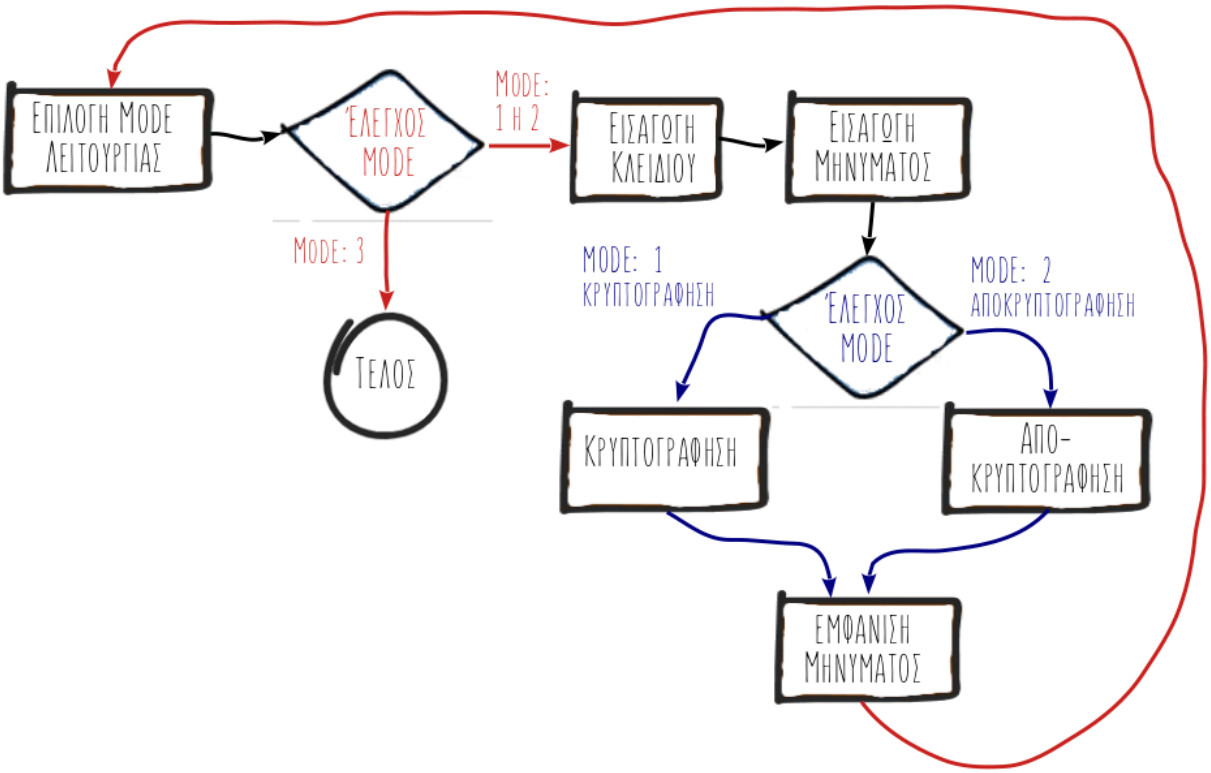
Η συνάρτηση αυτή αποκρυπτογραφεί ένα μήνυμα. Άρα χρειάζεται ως είσοδο (παράμετροι μέσα στην παρένθεση) δύο πράγματα: το κρυπτογραφημένο μήνυμα, το κλειδί.

Αυτό που επιστρέφει (εντολή return) είναι το αρχικό μήνυμα.

Κώδικας – Βήμα 5ο: Το κυρίως πρόγραμμα

Το κυρίως πρόγραμμα θα καλεί όλες τις παραπάνω functions στα σωστά σημεία και με τις σωστές παραμέτρους.

Θα έχει επαναληπτική δομή και θα τερματίζεται όταν ο χρήστης επιλέξει mode 3.



Στο παραπάνω διάγραμμα βλέπουμε ότι σε δύο σημεία του προγράμματος γίνεται έλεγχος του mode λειτουργίας (κόκκινος και μπλε ρόμβος).

Ποιές είναι οι διαφορές μεταξύ των δύο αυτών ελέγχων (ως προς το τι ακριβώς ελέγχουν και ως προς το ποια είναι η επίδρασή τους την ροή του προγράμματος) ;

Με βάση το παραπάνω διάγραμμα δημιουργείστε τον κώδικα του κυρίως προγράμματος. Μπορείτε επίσης να χρησιμοποιήσετε τον πιο κάτω ημιτελή κώδικα.

```
# κυρίως πρόγραμμα

a = getMode()

while .....:

    if a == 1:
        text = input('Message: ')
        ..... = getKey()
        ..... = encrypt(...., .....)
        print('Encrypted message: ', .....)

    else:
        ..... = input('Encrypted message: ')
        ..... = getKey()
        ..... = decrypt(....., .....)
        print('Decrypted message: ', .....)

    a = .....

print('bye')
```

Το κυρίως πρόγραμμα θα καλεί όλες τις παραπάνω functions στα σωστά σημεία και με τις σωστές παραμέτρους.

Θα έχει επαναληπτική δομή και θα τερματίζεται όταν ο χρήστης επιλέξει mode 3.

Ασκήσεις

1. Να τροποποιήσετε το πρόγραμμά σας, ώστε να κρυπτογραφούνται μόνο τα γράμματα. Οι αριθμοί και τα σύμβολα στίξης να μην αλλάζουν αλλάζουν.

Δοκιμάστε αρχικά στο IDLE την εντολή isalpha():

```
>>> symbol = 'a'
>>> symbol.isalpha()

>>> symbol = '5'
>>> symbol.isalpha()

>>> symbol = '!'
>>> symbol.isalpha()
```

2. Να τροποποιήσετε πρόγραμμά σας ώστε να αποκρυπτογραφεί οποιοδήποτε μήνυμα χωρίς να γνωρίζει το κλειδί, με την μέθοδο brute-force.

Η μέθοδος brute-force σημαίνει: δοκιμάζουμε όλα τα δυνατά κλειδιά μέχρις ότου το αποκρυπτογραφημένο μήνυμα να βγάζει νόημα.