



Girls' Programming Network

Cryptography

P

*Create a Vigenère Cipher encryptor and
decryptor!*

TUTORS ONLY



Part 1: Vigenère Ciphers

Task 1.1: Hide this message

Encrypt “can you keep this hidden” using the key “code”.
We’ve done some of the work for you. .

c	a	n	y	o	u	k	e	e	p	t	h	i	s	h	i	d	d	e	n
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
c	o	d	e	c	o	d	e	c	o	d	e	c	o	d	e	c	o	d	e
2	14	3	4	2	14	3	4	2	14	3	4	2	14	3	4	2	14	3	4
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
e	o	q	c	q	i	n	i	q	d	w	l	k	q	k	m	f	r	h	r

Task 2.2: Hide another

Encrypt “hide this message” using the key “key”.

h	i	d	e	t	h	i	s	m	e	s	s	a	g	e
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
k	e	y	k	e	y	k	e	y	k	e	y	k	e	y
10	4	24	10	4	24	10	4	24	10	4	24	10	4	24
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
r	m	b	o	x	f	s	w	k	o	w	q	k	k	c

Task 1.3: Uncover the message

Can you **decrypt** this message using the key “hack”.
We’ve done some of the work for you. .

f	o	w	m	y	a	e	u	l	d	v	r	l	c	q	n	l
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
h	a	c	k	h	a	c	k	h	a	c	k	h	a	c	k	h
-7	0	-2	-10	-7	0	-2	-10	-7	0	-2	-10	-7	0	-2	-10	-7
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
y	o	u	c	r	a	c	k	e	d	t	h	e	c	o	d	e

Task 1.4: Crack another

Can you **decrypt** this message using the key “**bug**”.

u	b	k		f	u	m	m	y		n	b	m		r	b	h	j	f	x
↓	↓	↓		↓	↓	↓	↓	↓		↓	↓	↓		↓	↓	↓	↓	↓	↓
b	u	q		b	u	q	b	u		q	b	u		q	b	u	q	b	u
-1	-20	-6		-1	-20	-6	-1	-20		-6	-1	-20		-6	-1	-20	-6	-1	-20
↓	↓	↓		↓	↓	↓	↓	↓		↓	↓	↓		↓	↓	↓	↓	↓	↓
<u>t</u>	<u>h</u>	<u>e</u>		<u>e</u>	<u>a</u>	<u>q</u>	<u>!</u>	<u>e</u>		<u>h</u>	<u>a</u>	<u>s</u>		<u>!</u>	<u>a</u>	<u>n</u>	<u>d</u>	<u>e</u>	<u>d</u>

Part 1: Key questions

Solution

The code should look like this (no bonuses):

```
# <the student's name>
print('Welcome to Vigenere Cipher')
message = input('What do you want to encrypt? ')
key = input('What is the encryption key? ')
```

Bonuses: Solution

The code should look like this (with bonuses):

```
# <the student's name>
name = input('What is your name? ')
print('Welcome to the Vigenere Cipher, ' + name)
message = input('What do you want to encrypt? ')
key = input('What is the encryption key? ')
```

Part 2: Calculating Keys!

Task 2.2: Create the alphabet

TUTOR TIPS

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
```

Encourage them to put their alphabet variable at the top of their code. This is the standard place to put global variables like this. It will make sure it's available when they need to use it in the code rather than it getting lost and in the wrong place in their code.

TUTOR TIPS

You can find the index of a character in a string by using `.index` like this:

```
alphabet.index(current_key_letter)
```

Full code Lesson 2

The code should look like this:

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
key_nums = []

name = input('What is your name? ')
print('Welcome to the Vigenere Cipher, ' + name)
message = input('What do you want to encrypt? ')
key = input('What is the encryption key? ')

for current_key_letter in key:
    current_key_num = alphabet.index(current_key_letter)
    key_nums.append(current_key_num)
print(key_nums)
```

Part 3: Encrypt!

Full code Lesson 3

Test of their function with parameters ('a',1) should return (and print) 'b' - a shifted by 1 is b

The code should look like this:

```
# <the student's name>
alphabet = 'abcdefghijklmnopqrstuvwxyz'
key_nums = []

def encrypt_letter(letter, key_num):
    current_index = alphabet.index(letter)
    new_index = (current_index + key_num) % 26
    new_letter = alphabet[new_index]
    return new_letter

name = input('What is your name? ')
print('Welcome to the Vigenere Cipher, ' + name)
message = input('What do you want to encrypt? ')
key = input('What is the encryption key? ')

for current_key_letter in key:
    current_key_num = alphabet.index(current_key_letter)
    key_nums.append(current_key_num)

result = encrypt_letter('a', 1)
print(result)
```

Part 4: Matching messages to keys

Task 4.5: Getting the right key

TUTOR TIPS

This is probably one of the hardest parts of the workbook both logically and in terms of python knowledge.

It has a complex combination of numbers and indexes.
We are looking for these lines:

```
# finds the position in the key number list that we will need
key_position = count % key_len

# Goes and gets the key number out of the list
current_key_num = key_nums[key_position]
```

Full code Lesson 4

The code should look like this:

```
# <the student's name>
alphabet = 'abcdefghijklmnopqrstuvwxyz'
key_nums = []

def encrypt_letter(letter, key_num):
    current_index = alphabet.index(letter)
    new_index = (current_index + key_num) % 26
    new_letter = alphabet[new_index]
    return new_letter

name = input('What is your name? ')
print('Welcome to the Vigenere Cipher, ' + name)
message = input('What do you want to encrypt? ')
key = input('What is the encryption key? ')

for current_key_letter in key:
    current_key_num = alphabet.index(current_key_letter)
    key_nums.append(current_key_num)

count = 0
key_len = len(key)

for current_letter in message:
    key_position = count % key_len
    current_key_num = key_nums[key_position]

    result = encrypt_letter(current_letter, current_key_num)
    print(result, end='')

    count += 1
    # print(key_position)
```

Part 5: Space Ace

Full code Lesson 5

The code should look like this:

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
key_nums = []

def encrypt_letter(letter, key_num):
    current_index = alphabet.index(letter)
    new_index = (current_index + key_num) % 26
    new_letter = alphabet[new_index]
    return new_letter

name = input('What is your name? ')
print('Welcome to the Vigenere Cipher, ' + name)
message = input('What do you want to encrypt? ')
key = input('What is the encryption key? ')

for current_key_letter in key:
    current_key_num = alphabet.index(current_key_letter)
    key_nums.append(current_key_num)

count = 0
key_len = len(key)

for current_letter in message:
    if current_letter in alphabet:
        key_position = count % key_len
        current_key_num = key_nums[key_position]

        result = encrypt_letter(current_letter,
current_key_num)
        print(result, end='')
        count += 1
    else:
        print(current_letter, end='')
```


Part 6: Decrypt!

TUTOR TIPS

Students may forget to use == in if statements

The code should look like this:

```
# <the student's name>
alphabet = 'abcdefghijklmnopqrstuvwxyz'
key_nums = []

def encrypt_letter(letter, key_num):
    current_index = alphabet.index(letter)
    new_index = (current_index + key_num) % 26
    new_letter = alphabet[new_index]
    return new_letter

name = input('What is your name? ')
print('Welcome to the Vigenere Cipher, ' + name)
mode = input('Do you want to encrypt or decrypt - e or d? ')
# there are many ways they can change the messages depending on if
# e or d entered
if mode == 'e':
    message = input('What do you want to encrypt? ')
    key = input('What is the encryption key? ')
else:
    message = input('What do you want to decrypt? ')
    key = input('What is the decryption key? ')

for current_key_letter in key:
    current_key_num = alphabet.index(current_key_letter)
    if mode == 'd':
        current_key_num *= -1
    key_nums.append(current_key_num)

count = 0
key_len = len(key)

for current_letter in message:
    if current_letter in alphabet:
        key_position = count % key_len
        current_key_num = key_nums[key_position]

        result = encrypt_letter(current_letter, current_key_num)
        print(result, end='')

        count += 1
    else:
        print(current_letter, end='')

```

Extension 7: Files!

TUTOR TIPS

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
key_nums = []

def encrypt_letter(letter, key_num):
    current_index = alphabet.index(letter)
    new_index = (current_index + key_num) % 26
    new_letter = alphabet[new_index]
    return new_letter

name = input('What is your name? ')
print('Welcome to the Vigenere Cipher, ' + name)
mode = input('Do you want to encrypt or decrypt - e or d? ')
if mode == 'e':
    key = input('What is the encryption key? ')
else:
    key = input('What is the decryption key? ')

f = open('message.txt', 'r')
message = f.read()

for current_key_letter in key:
    current_key_num = alphabet.index(current_key_letter)
    if mode == 'decrypt':
        current_key_num *= -1
    key_nums.append(current_key_num)
count = 0
key_len = len(key)

for current_letter in message:
    if current_letter in alphabet:
        key_position = count % key_len
        current_key_num = key_nums[key_position]

        result = encrypt_letter(current_letter, current_key_num)
        print(result, end='')

        count += 1
    else:
        print(current_letter, end='')
```