

Fundamentals of Computer Programming

Building a Programming Portfolio

Week 6

You should be able to complete the following programs by the end of the week. By now you should understand why you should be saving your work to GitHub or similar. Possible solutions will be uploaded to the main module GitHub repository every week. If you follow that repo you should be able to receive notifications.

Note that as we have now seen many features of Python there are often multiple ways to solve a problem. One lesson this week should be that a list can often be used in place of a tuple, and vice versa. The aim is to go for the most obvious, and most clear, solution.

1. Write a function that accepts a positive integer as a parameter and then returns a representation of that number in binary (base 2).



```
3
4
5 def binary(num):
6     if num<0:
7         print("The value must be positive")
8     num_in_binary=bin(num)[2:]
9     return num_in_binary
10
11 print(binary(10))
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ...

```
PS C:\Users\user> python -u "d:\3rd Semester\FOCP\28th december\binary.py"
1010
PS C:\Users\user>
```

Hint: This is in many ways a trick question. Think!

2. Write and test a function that takes an integer as its parameter and returns the *factors* of that integer. (A factor is an integer which can be multiplied by another to yield the original).

```

19 def factors(num):
20     factors_list = []
21     for i in range(1, num + 1):
22         if num%i==0:
23             factors_list.append(i)
24
25     return factors_list
26
27 f=factors(16)
28 print("The factors of number 16 are", f)
29

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

```

PS C:\Users\user> python -u "d:\3rd Semester\F0CP\28th december\tempCodeRunnerFile.py"
The factors of number 16 are [1, 2, 4, 8, 16]
PS C:\Users\user>

```

3. Write and test a function that determines if a given integer is a prime number. A prime number is an integer greater than 1 that cannot be produced by multiplying two other integers.

```

1 num=int(input("Enter a number:"))
2 if num==1:
3     print("It is not a prime number")
4 if num>1:
5     for i in range(2,num):
6         if num % i==0:
7             print("It is not a prime number")
8             break
9         else:
10            print("It is a prime number")
11            break

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\3rd Semester\F0CP\28th december> python -u "d:\3rd Semester\F0CP\28th december\tempCodeRunnerFile.py"
Enter a number:17
It is a prime number
PS D:\3rd Semester\F0CP\28th december>

```

4. Computers are commonly used in encryption. A very simple form of encryption (more accurately "obfuscation") would be to remove the spaces from a message and reverse the resulting string. Write, and test, a function that takes a string containing a message and "encrypts" it in this way.

```

13 string="This message needs to be encrypted"
14 encryption=string.replace(" ", "")[::-1]
15 print(encryption)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\3rd Semester\FOCP\28th december> python -u "d:\3rd Semester\FOCP\28th december\tempCodeRunnerFile.py"
detpyrcneebotsdeenegassemihT
PS D:\3rd Semester\FOCP\28th december>

```

5. Another way to hide a message is to include the letters that make it up within seemingly random text. The letters of the message might be every fifth character, for example. Write and test a function that does such encryption. It should randomly generate an interval (between 2 and 20), space the message out accordingly, and should fill the gaps with random letters. The function should return the encrypted message and the interval used. For example, if the message is "send cheese", the random interval is 2, and for clarity the random letters are not random:

```

send cheese
s e n d c h e e s e
sxyexynxydxy cxyhxyexyexysxye

```

```

17 import random
18 import string
19
20 def encrypt_message(message):
21     interval = random.randint(2, 20)
22
23     encrypted_message = []
24
25     for char in message:
26         encrypted_message.append(char)
27
28         for _ in range(interval - 1):
29             random_letter = random.choice(string.ascii_lowercase)
30             encrypted_message.append(random_letter)
31
32     encrypted_message_str = ''.join(encrypted_message)
33
34     return encrypted_message_str, interval
35
36 if __name__ == "__main__":
37     message = "send cheese"
38     encrypted_message, interval = encrypt_message(message)
39     print(f"Original message: {message}")
40     print(f"Encrypted message: {encrypted_message}")
41     print(f"Interval used: {interval}")
42

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\3rd Semester\FOCP\28th december> python -u "d:\3rd Semester\FOCP\28th december\tempCodeRunnerFile.py"
Original message: send cheese
Encrypted message: sfelnrdy ncbhqememsxel
Interval used: 2
PS D:\3rd Semester\FOCP\28th december>

```

6. Write a program that decrypts messages encoded as above.

```
17 import random
18 import string
19
20 def encrypt_message(message):
21     interval = random.randint(2, 20)
22
23     encrypted_message = []
24
25     for char in message:
26         encrypted_message.append(char)
27
28         for _ in range(interval - 1):
29             random_letter = random.choice(string.ascii_lowercase)
30             encrypted_message.append(random_letter)
31
32     encrypted_message_str = ''.join(encrypted_message)
33
34     return encrypted_message_str, interval
35
36 if __name__ == "__main__":
37     message = "send cheese"
38     encrypted_message, interval = encrypt_message(message)
39
40 def decrypt_message(encrypted_message, interval):
41     original_message = []
42
43     for i in range(0, len(encrypted_message), interval):
44         original_message.append(encrypted_message[i])
45
46     return ''.join(original_message)
47
48 decrypted_message=decrypt_message(encrypted_message,interval)
49 print(decrypted_message)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\3rd Semester\FOCP\28th december> python -u "d:\3rd Semester\FOCP\28th december\tempCodeRunnerFile.py"
send cheese
PS D:\3rd Semester\FOCP\28th december> █
```