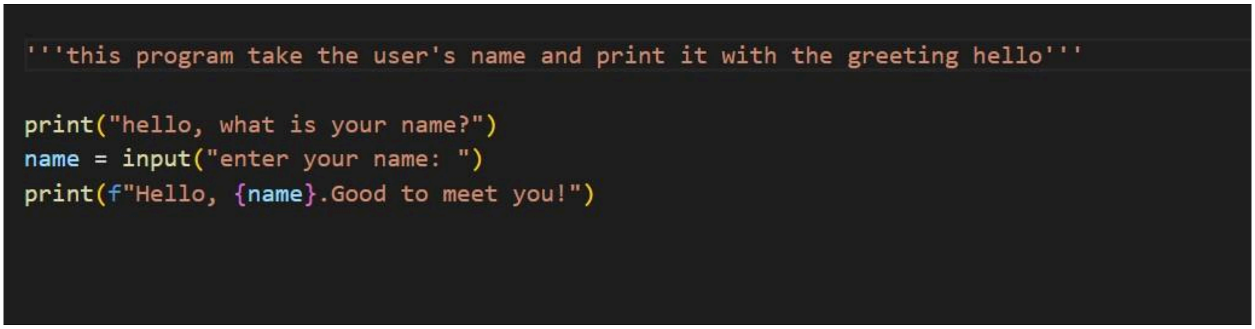
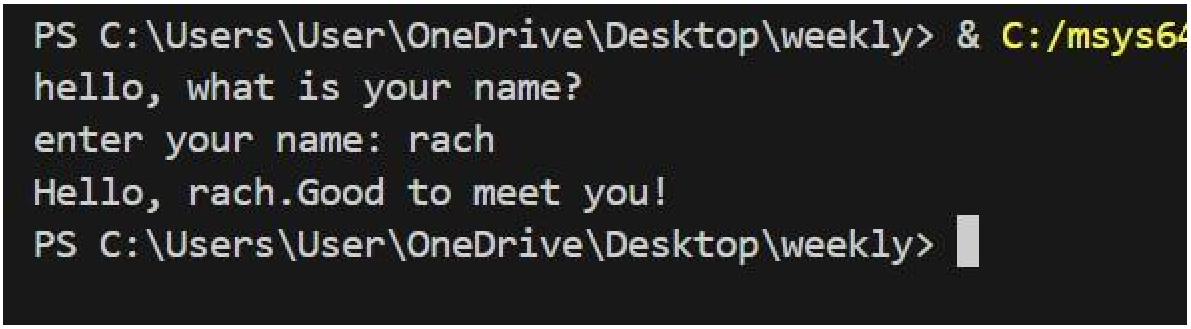
Documentation of weekly programs

Week-2

question1:

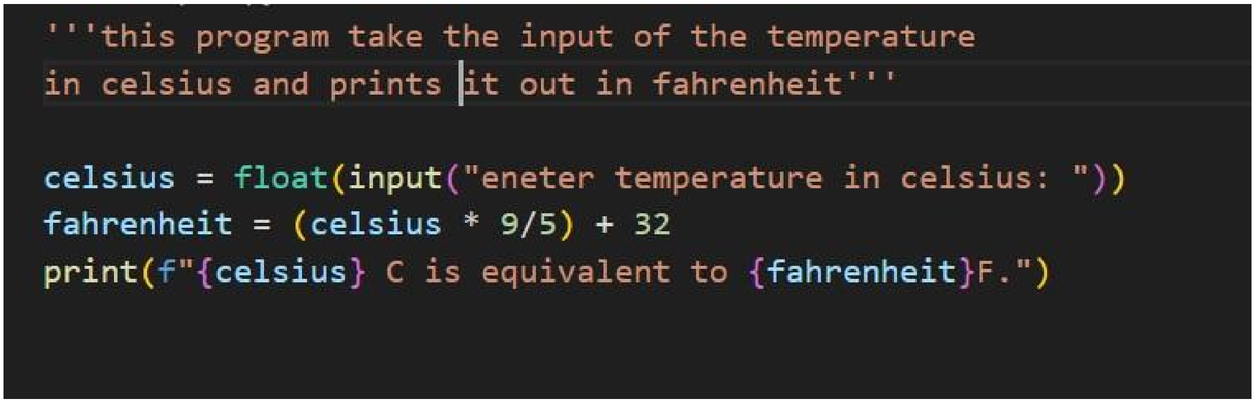


```python
'''this program take the user's name and print it with the greeting hello'''

print("hello, what is your name?")
name = input("enter your name: ")
print(f"Hello, {name}.Good to meet you!")
```

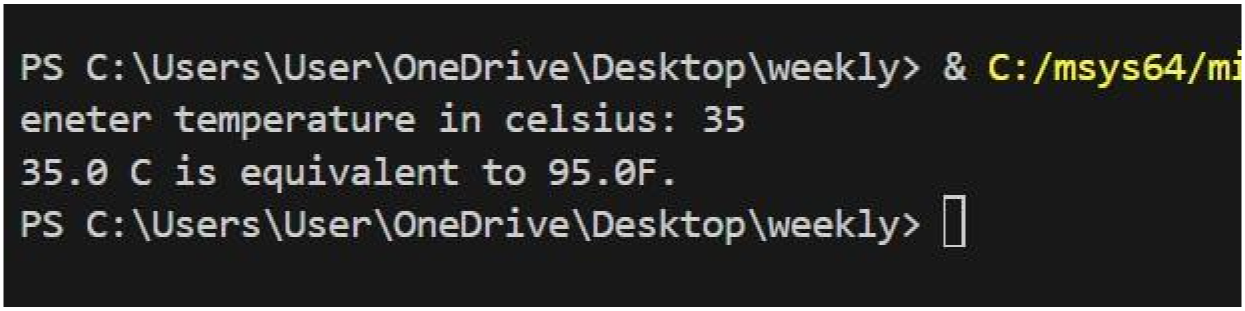


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64
hello, what is your name?
enter your name: rach
Hello, rach.Good to meet you!
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question2:

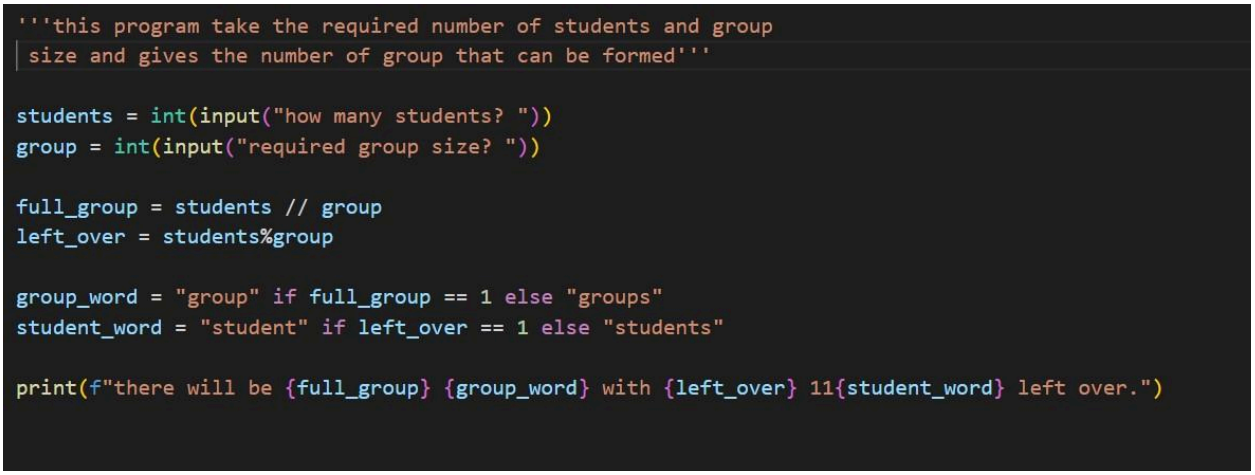


```
'''this program take the input of the temperature
in celsius and prints it out in fahrenheit'''

celsius = float(input("eneter temperature in celsius: "))
fahrenheit = (celsius * 9/5) + 32
print(f"{celsius} C is equivalent to {fahrenheit}F.")
```

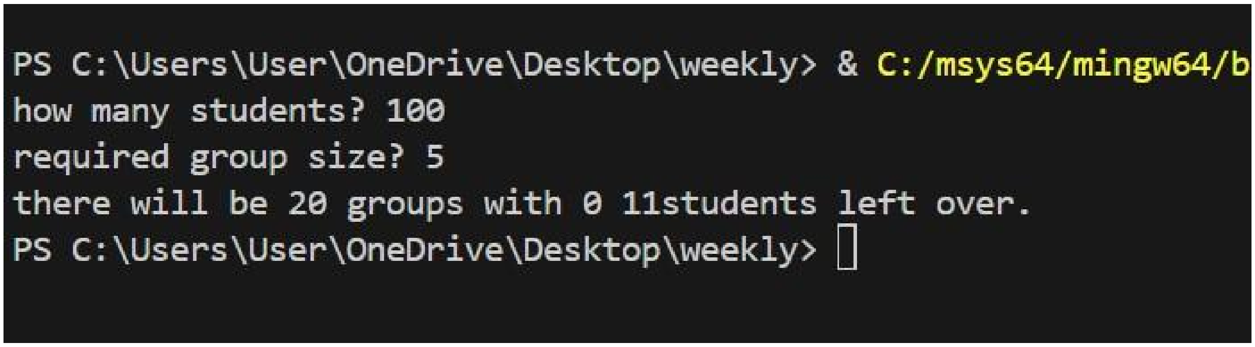


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mi
eneter temperature in celsius: 35
35.0 C is equivalent to 95.0F.
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 3:

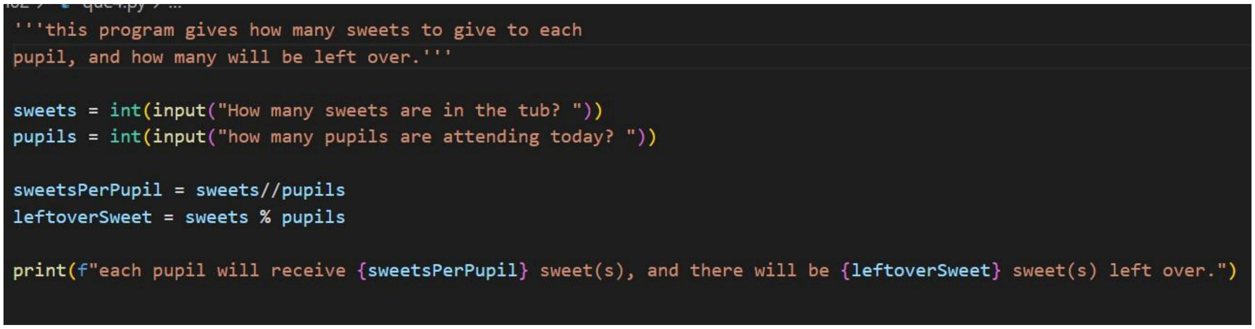


```python
'''this program take the required number of students and group
 size and gives the number of group that can be formed'''

students = int(input("how many students? "))
group = int(input("required group size? "))

full_group = students // group
left_over = students%group

group_word = "group" if full_group == 1 else "groups"
student_word = "student" if left_over == 1 else "students"

print(f"there will be {full_group} {group_word} with {left_over} 11{student_word} left over.")
```

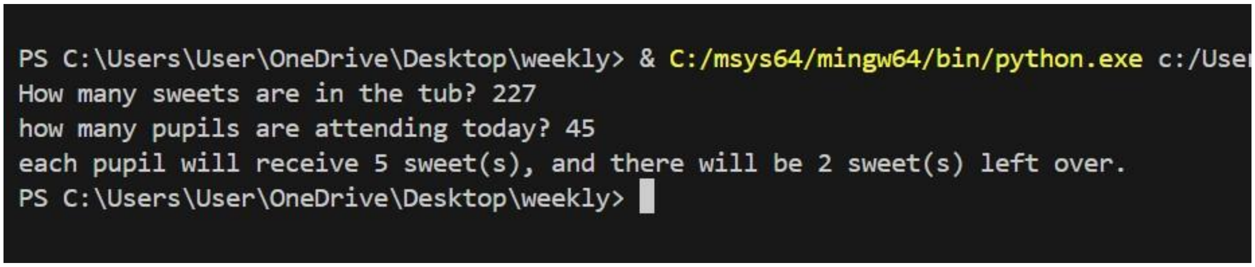


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/b
how many students? 100
required group size? 5
there will be 20 groups with 0 11students left over.
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 4:



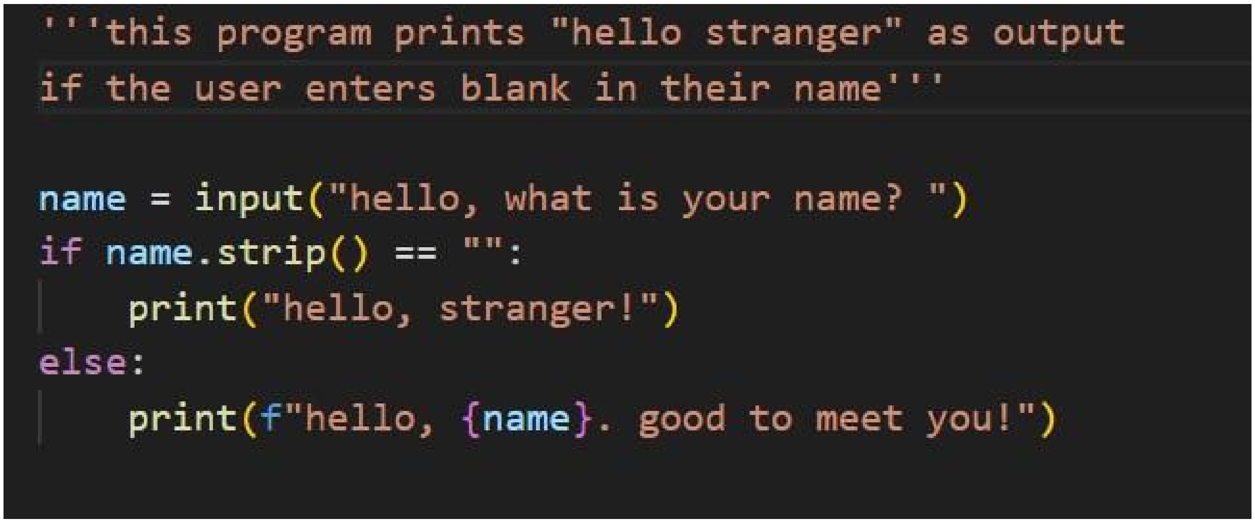
```python
'''this program gives how many sweets to give to each
pupil, and how many will be left over.'''

sweets = int(input("How many sweets are in the tub? "))
pupils = int(input("how many pupils are attending today? "))

sweetsPerPupil = sweets//pupils
leftoverSweet = sweets % pupils

print(f"each pupil will receive {sweetsPerPupil} sweet(s), and there will be {leftoverSweet} sweet(s) left over.")
```



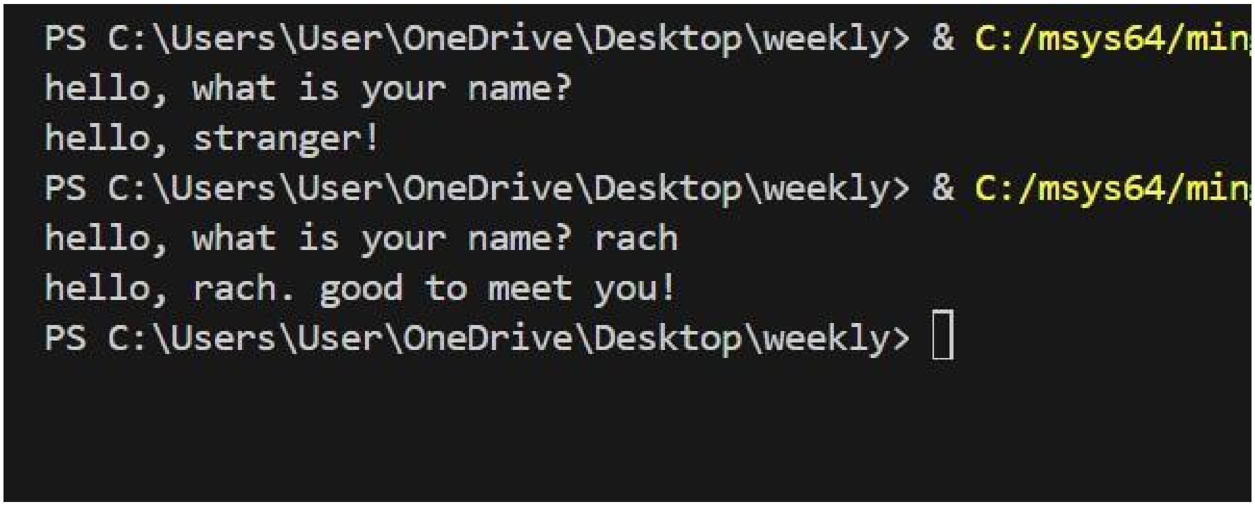
```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/python.exe c:/User
How many sweets are in the tub? 227
how many pupils are attending today? 45
each pupil will receive 5 sweet(s), and there will be 2 sweet(s) left over.
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Week-3

Question 1:

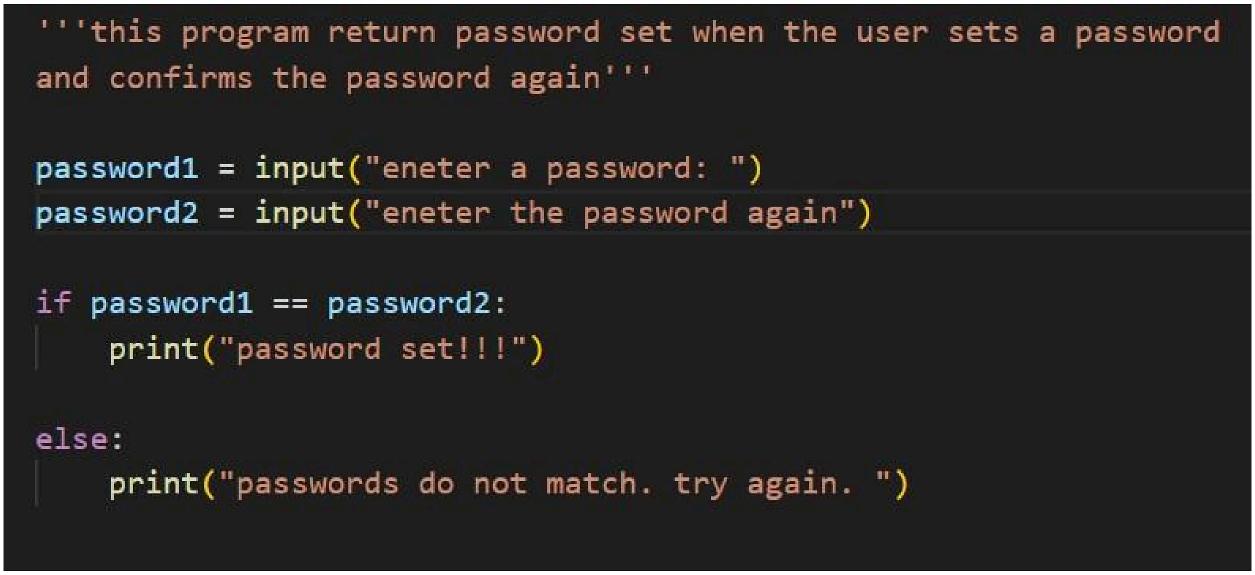


```python
'''this program prints "hello stranger" as output
if the user enters blank in their name'''

name = input("hello, what is your name? ")
if name.strip() == "":
    print("hello, stranger!")
else:
    print(f"hello, {name}. good to meet you!")
```

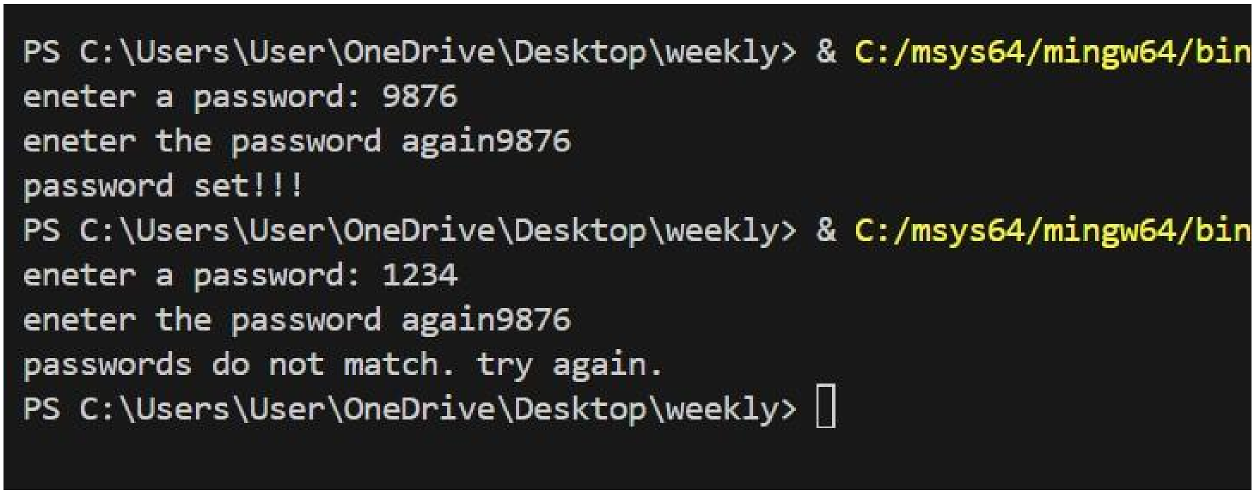


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/min
hello, what is your name?
hello, stranger!
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/min
hello, what is your name? rach
hello, rach. good to meet you!
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 2:

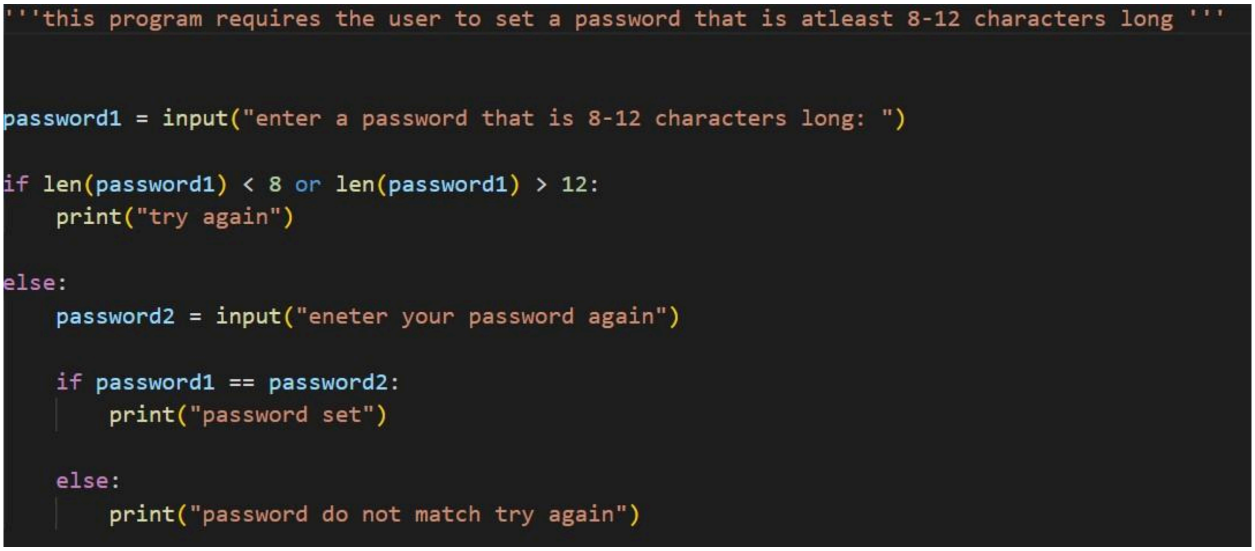


```python
'''this program return password set when the user sets a password
and confirms the password again'''

password1 = input("eneter a password: ")
password2 = input("eneter the password again")

if password1 == password2:
    print("password set!!!")

else:
    print("passwords do not match. try again. ")
```

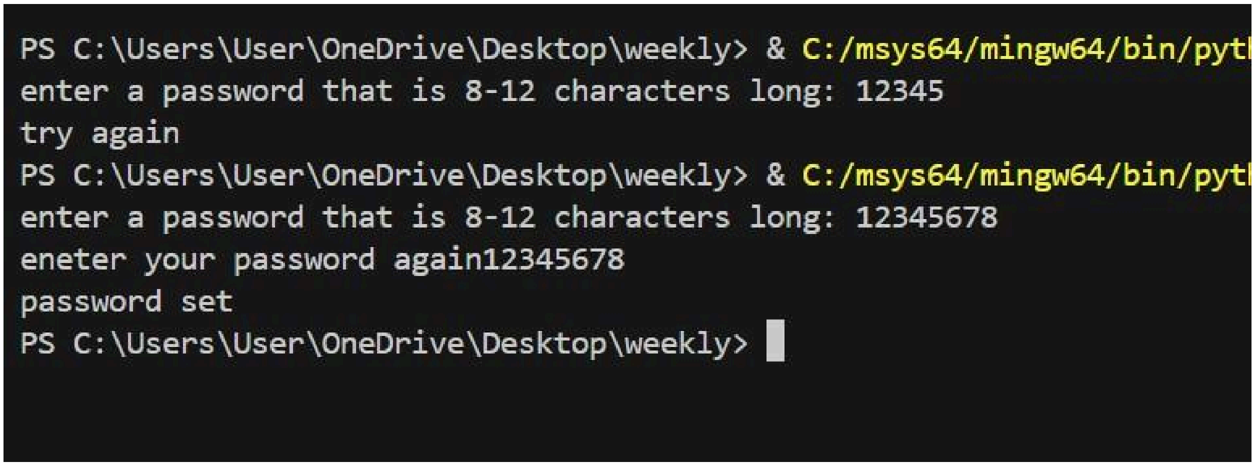


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin
eneter a password: 9876
eneter the password again9876
password set!!!
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin
eneter a password: 1234
eneter the password again9876
passwords do not match. try again.
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 3:

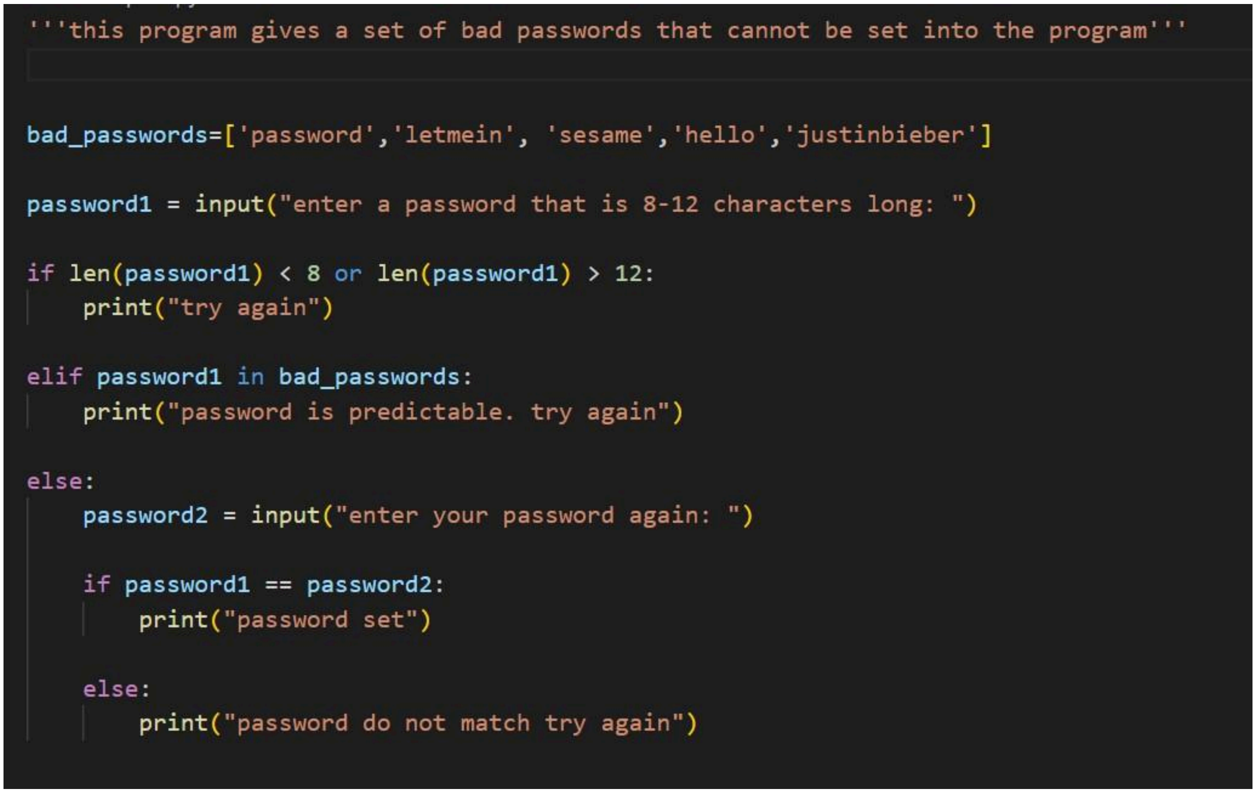


```
'''this program requires the user to set a password that is atleast 8-12 characters long '''


password1 = input("enter a password that is 8-12 characters long: ")

if len(password1) < 8 or len(password1) > 12:
    print("try again")

else:
    password2 = input("eneter your password again")

    if password1 == password2:
        print("password set")

    else:
        print("password do not match try again")
```

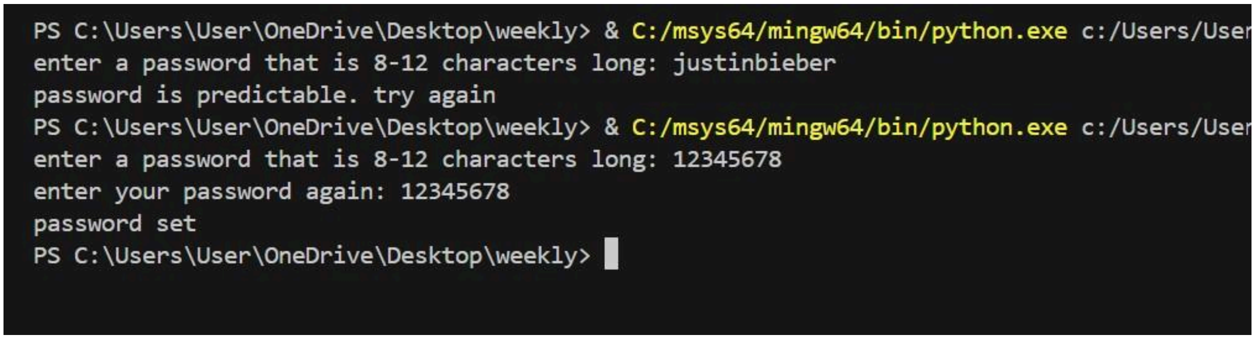


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/pyt
enter a password that is 8-12 characters long: 12345
try again
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/pyt
enter a password that is 8-12 characters long: 12345678
eneter your password again12345678
password set
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 4:

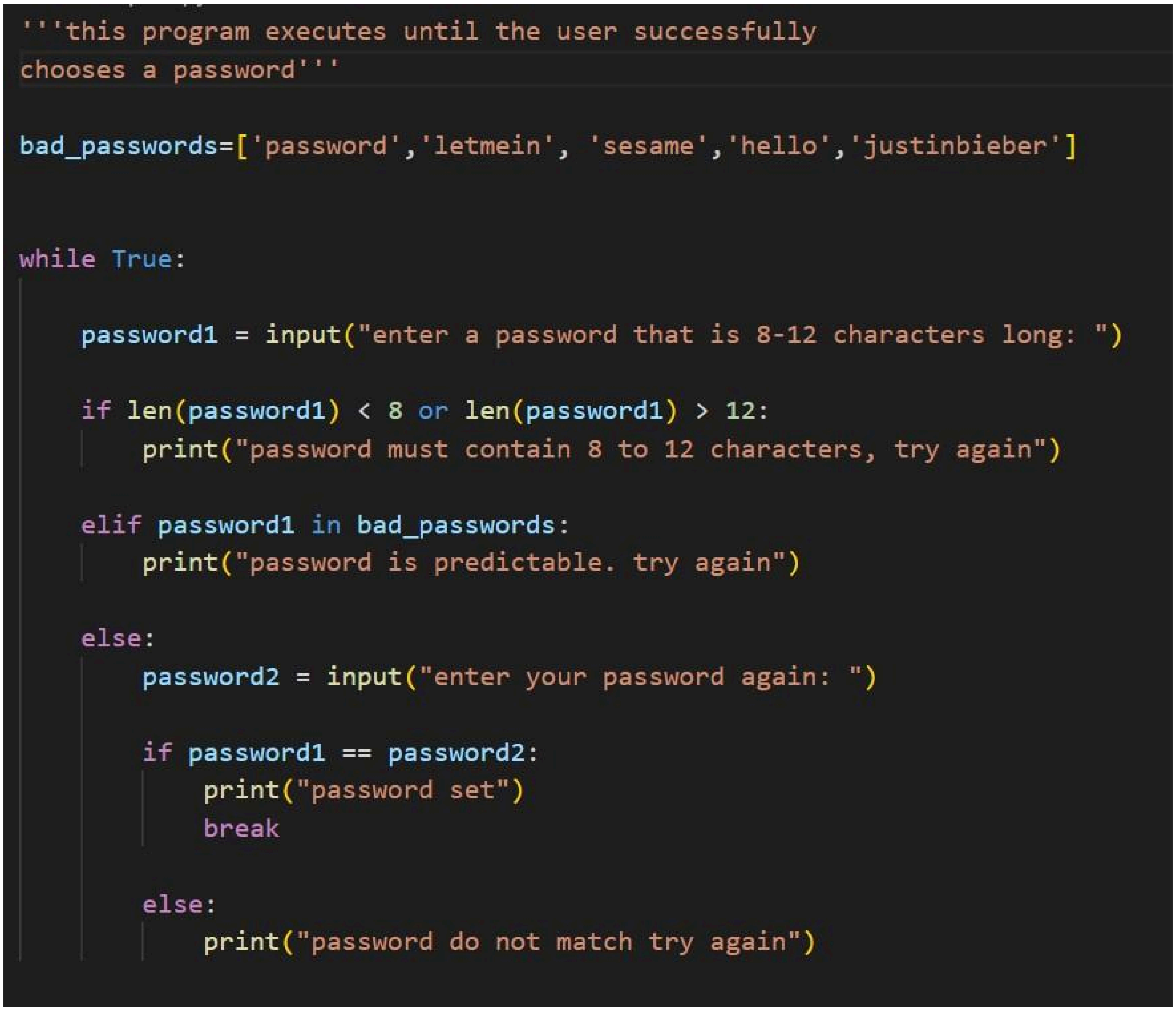


```python
'''this program gives a set of bad passwords that cannot be set into the program'''


bad_passwords=['password','letmein', 'sesame','hello','justinbieber']

password1 = input("enter a password that is 8-12 characters long: ")

if len(password1) < 8 or len(password1) > 12:
    print("try again")

elif password1 in bad_passwords:
    print("password is predictable. try again")

else:
    password2 = input("enter your password again: ")

    if password1 == password2:
        print("password set")

    else:
        print("password do not match try again")
```

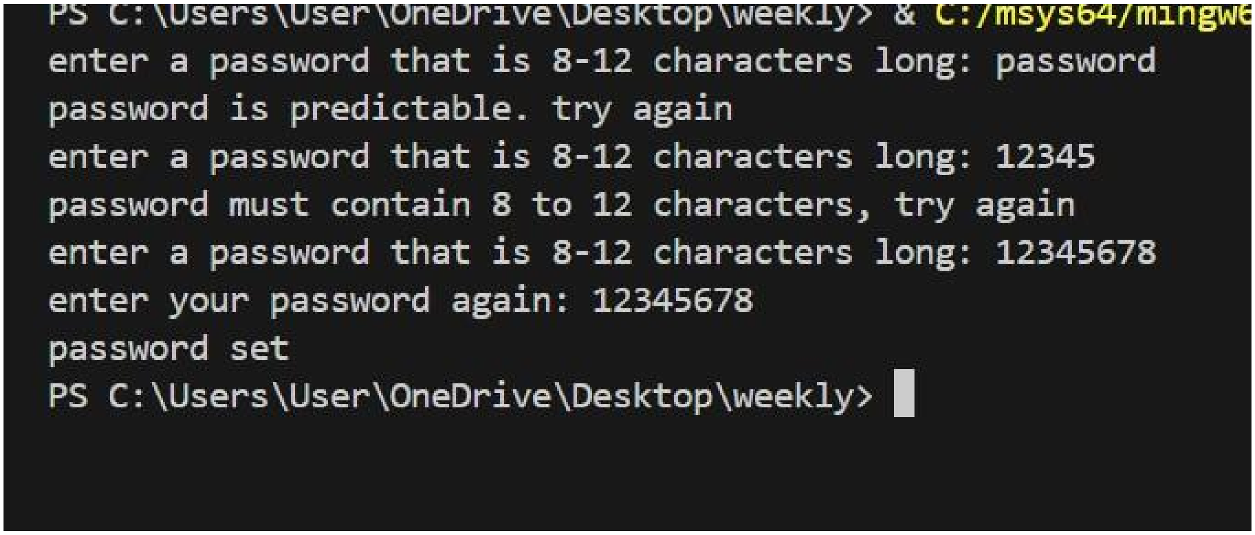


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/python.exe c:/Users/User
enter a password that is 8-12 characters long: justinbieber
password is predictable. try again
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/python.exe c:/Users/User
enter a password that is 8-12 characters long: 12345678
enter your password again: 12345678
password set
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 5:

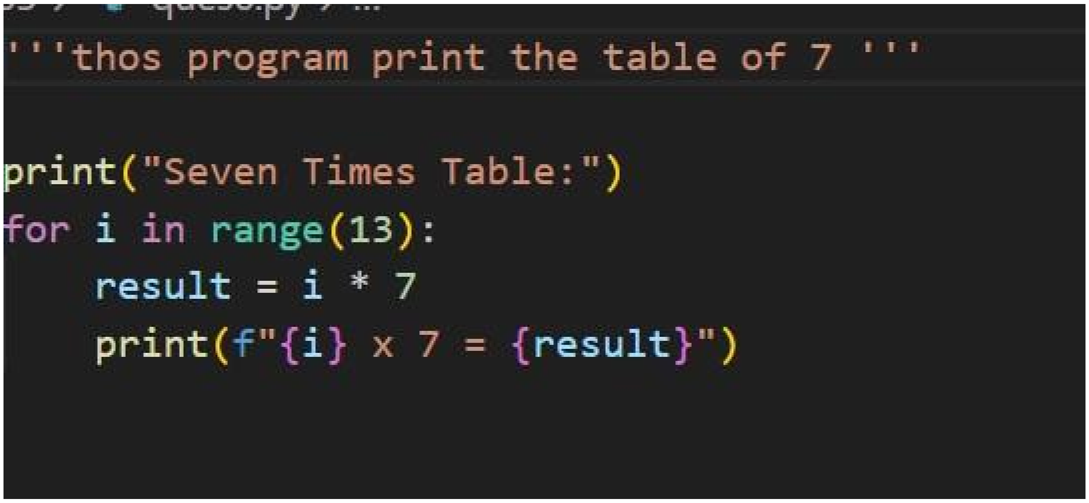


```
'''this program executes until the user successfully
chooses a password'''

bad_passwords=['password','letmein', 'sesame','hello','justinbieber']


while True:

    password1 = input("enter a password that is 8-12 characters long: ")

    if len(password1) < 8 or len(password1) > 12:
        print("password must contain 8 to 12 characters, try again")

    elif password1 in bad_passwords:
        print("password is predictable. try again")

    else:
        password2 = input("enter your password again: ")

        if password1 == password2:
            print("password set")
            break

        else:
            print("password do not match try again")
```

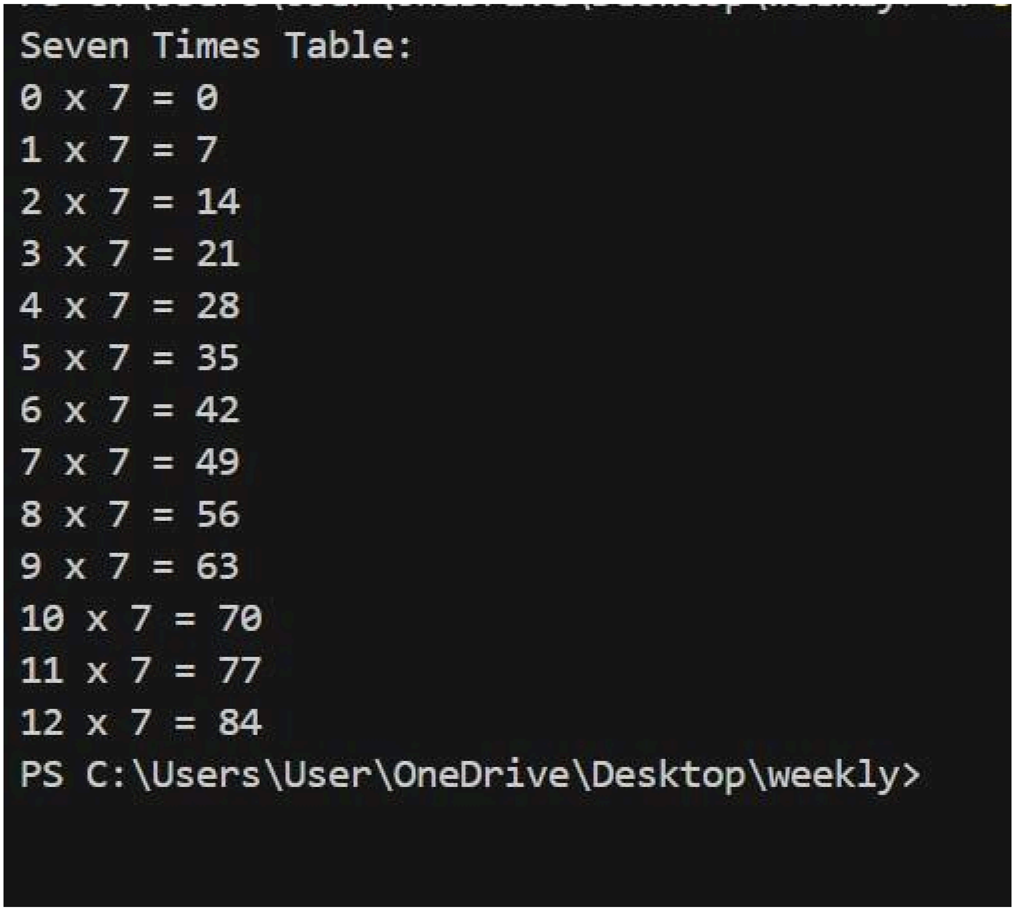


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw6
enter a password that is 8-12 characters long: password
password is predictable. try again
enter a password that is 8-12 characters long: 12345
password must contain 8 to 12 characters, try again
enter a password that is 8-12 characters long: 12345678
enter your password again: 12345678
password set
PS C:\Users\User\OneDrive\Desktop\weekly> █
```

Question 6:

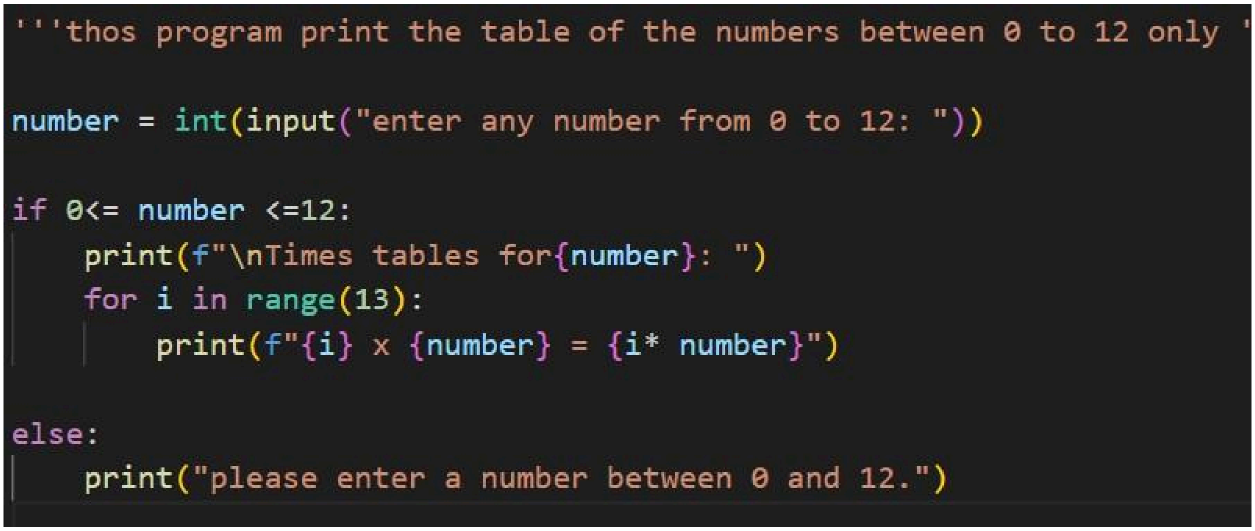


```
'''thos program print the table of 7 '''

print("Seven Times Table:")
for i in range(13):
    result = i * 7
    print(f"{i} x 7 = {result}")
```

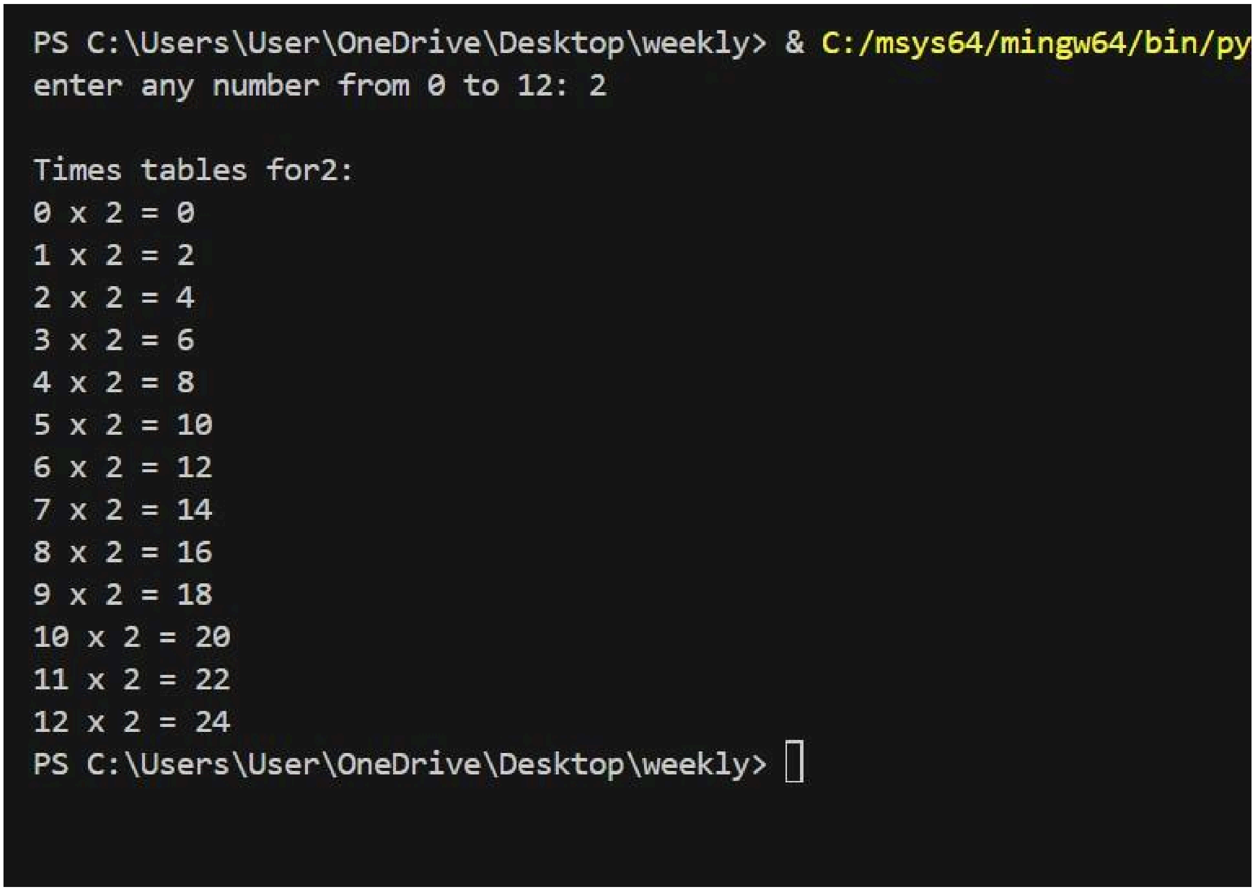


```
Seven Times Table:
0 x 7 = 0
1 x 7 = 7
2 x 7 = 14
3 x 7 = 21
4 x 7 = 28
5 x 7 = 35
6 x 7 = 42
7 x 7 = 49
8 x 7 = 56
9 x 7 = 63
10 x 7 = 70
11 x 7 = 77
12 x 7 = 84
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 7:

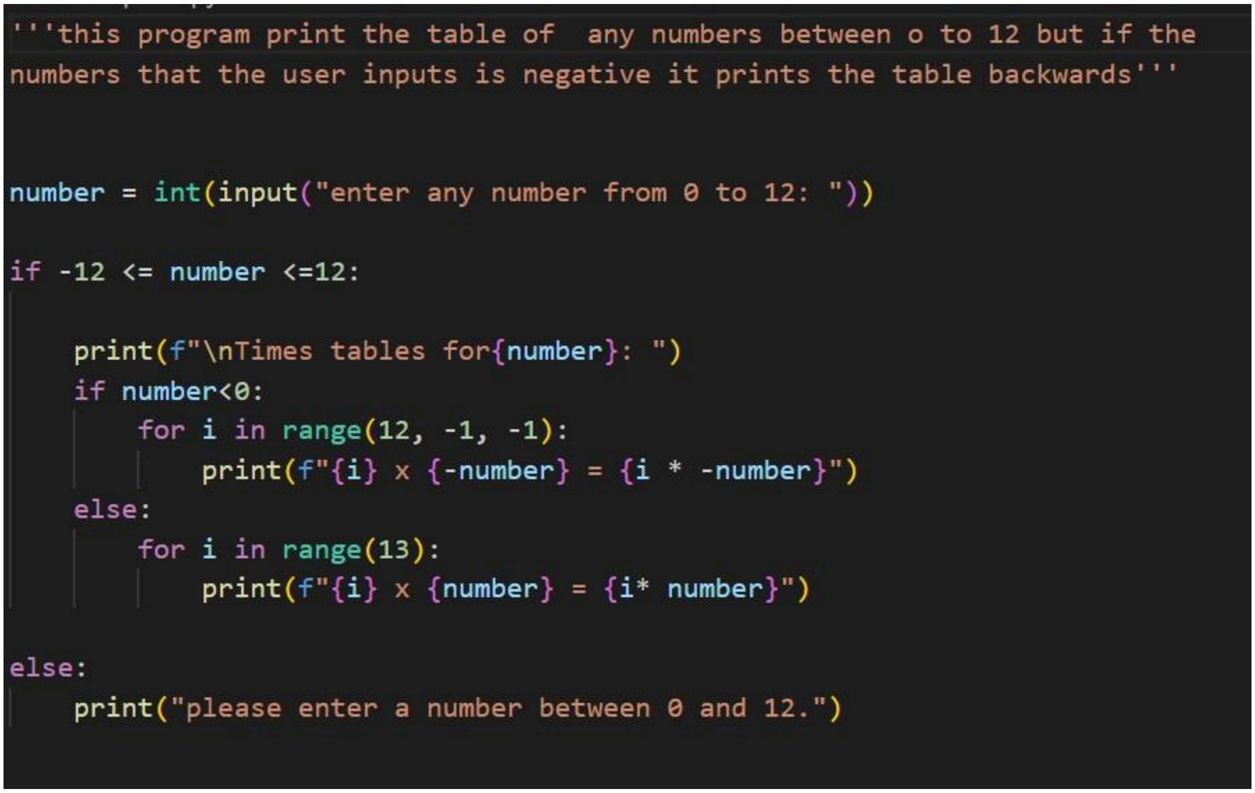


```
'''thos program print the table of the numbers between 0 to 12 only '

number = int(input("enter any number from 0 to 12: "))

if 0<= number <=12:
    print(f"\nTimes tables for{number}: ")
    for i in range(13):
        print(f"{i} x {number} = {i* number}")

else:
    print("please enter a number between 0 and 12.")
```

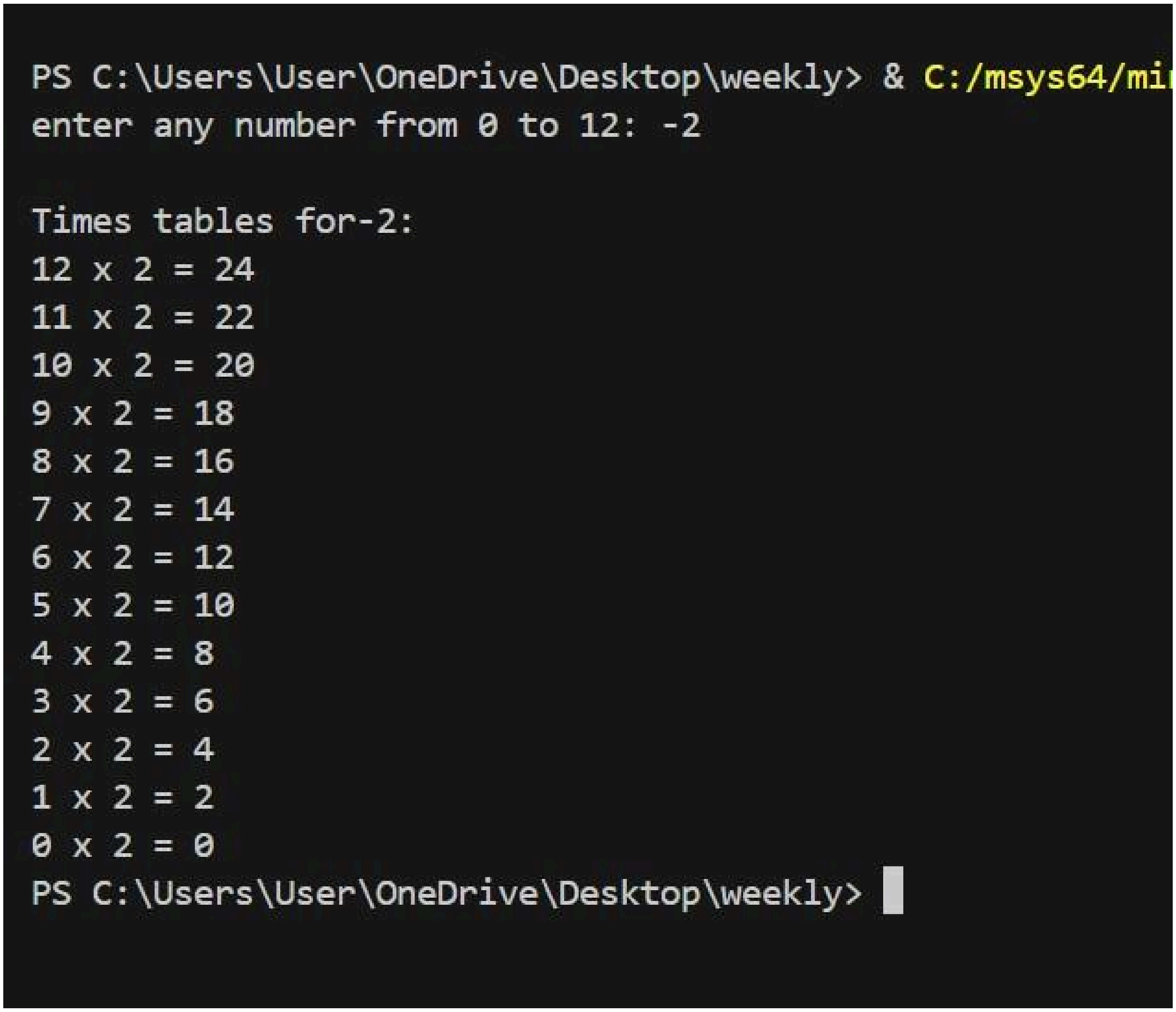


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/py
enter any number from 0 to 12: 2

Times tables for2:
0 x 2 = 0
1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
4 x 2 = 8
5 x 2 = 10
6 x 2 = 12
7 x 2 = 14
8 x 2 = 16
9 x 2 = 18
10 x 2 = 20
11 x 2 = 22
12 x 2 = 24
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 8:



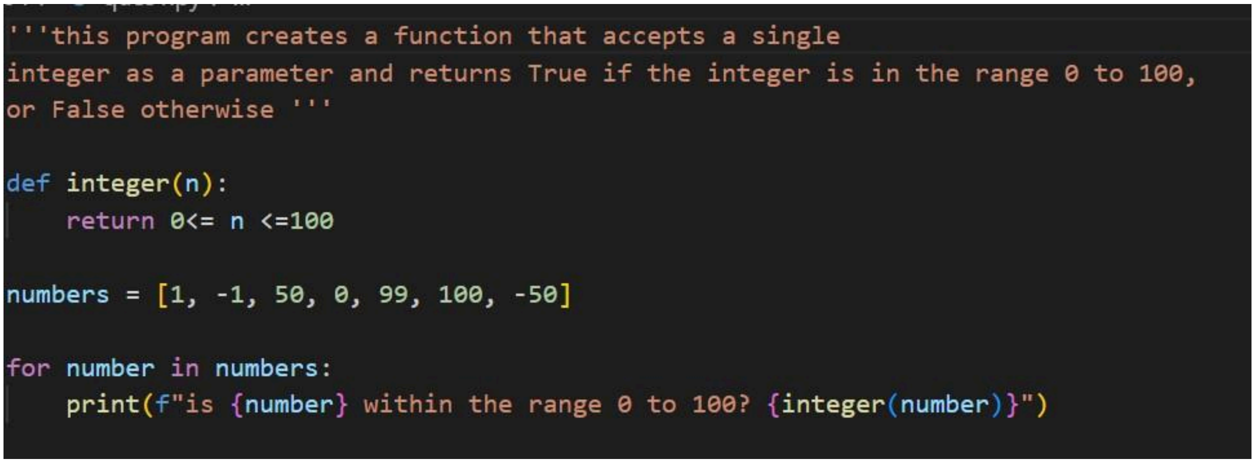
```
'''this program print the table of  any numbers between o to 12 but if the
numbers that the user inputs is negative it prints the table backwards'''


number = int(input("enter any number from 0 to 12: "))

if -12 <= number <=12:

    print(f"\nTimes tables for{number}: ")
    if number<0:
        for i in range(12, -1, -1):
            print(f"{i} x {-number} = {i * -number}")
    else:
        for i in range(13):
            print(f"{i} x {number} = {i* number}")

else:
    print("please enter a number between 0 and 12.")
```



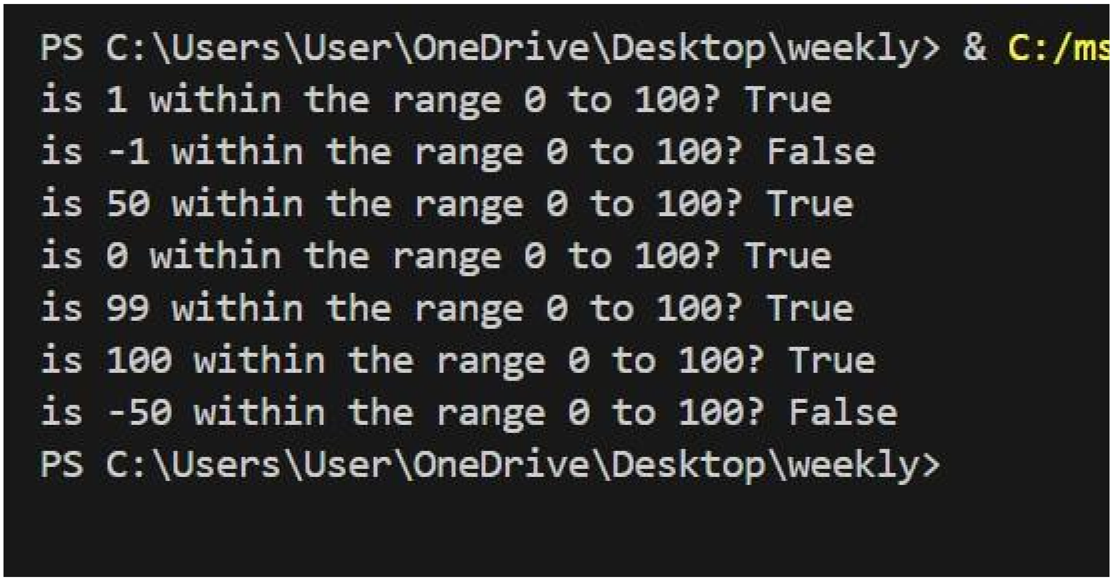
```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mi
enter any number from 0 to 12: -2

Times tables for-2:
12 x 2 = 24
11 x 2 = 22
10 x 2 = 20
9 x 2 = 18
8 x 2 = 16
7 x 2 = 14
6 x 2 = 12
5 x 2 = 10
4 x 2 = 8
3 x 2 = 6
2 x 2 = 4
1 x 2 = 2
0 x 2 = 0
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Week -4

Question 1:

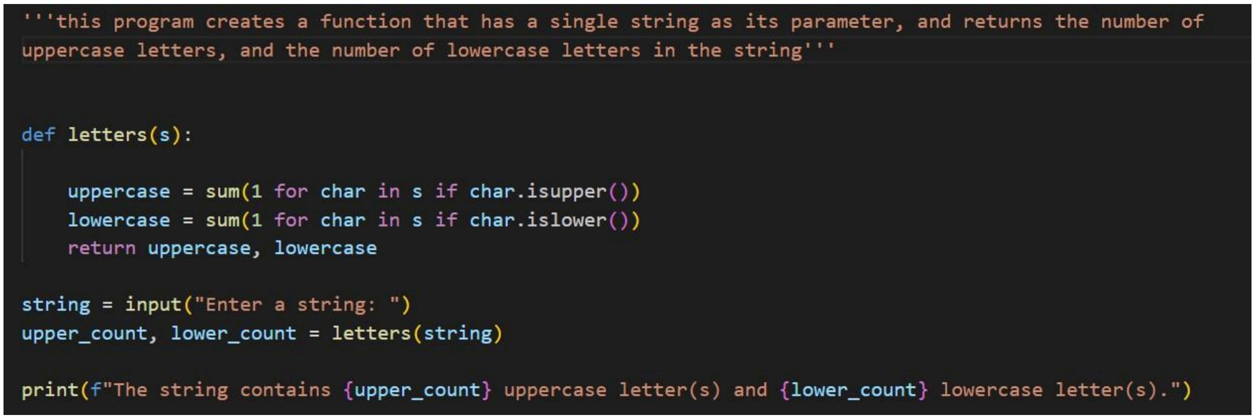


```
'''this program creates a function that accepts a single
integer as a parameter and returns True if the integer is in the range 0 to 100,
or False otherwise '''

def integer(n):
    return 0<= n <=100

numbers = [1, -1, 50, 0, 99, 100, -50]

for number in numbers:
    print(f"is {number} within the range 0 to 100? {integer(number)}")
```

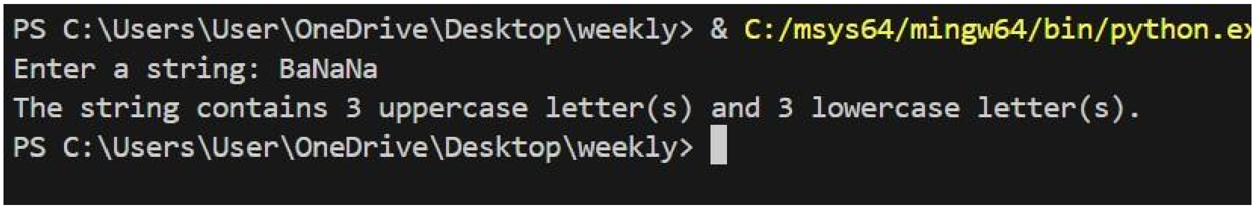


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/ms
is 1 within the range 0 to 100? True
is -1 within the range 0 to 100? False
is 50 within the range 0 to 100? True
is 0 within the range 0 to 100? True
is 99 within the range 0 to 100? True
is 100 within the range 0 to 100? True
is -50 within the range 0 to 100? False
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 2:

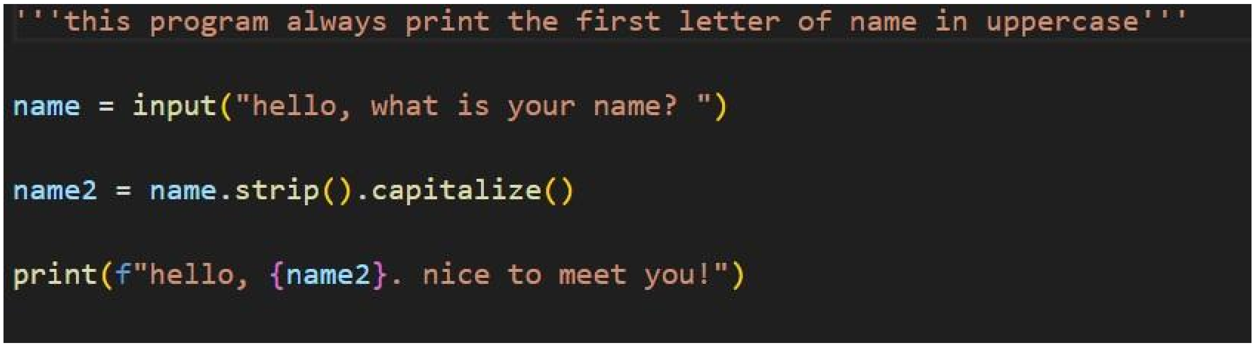


```python
'''this program creates a function that has a single string as its parameter, and returns the number of
uppercase letters, and the number of lowercase letters in the string'''


def letters(s):

    uppercase = sum(1 for char in s if char.isupper())
    lowercase = sum(1 for char in s if char.islower())
    return uppercase, lowercase

string = input("Enter a string: ")
upper_count, lower_count = letters(string)

print(f"The string contains {upper_count} uppercase letter(s) and {lower_count} lowercase letter(s).")
```

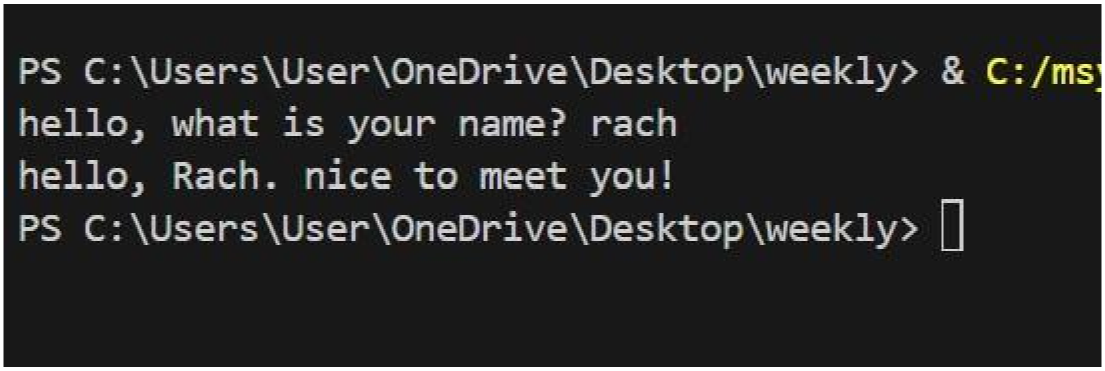


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/python.ex
Enter a string: BaNaNa
The string contains 3 uppercase letter(s) and 3 lowercase letter(s).
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 3:

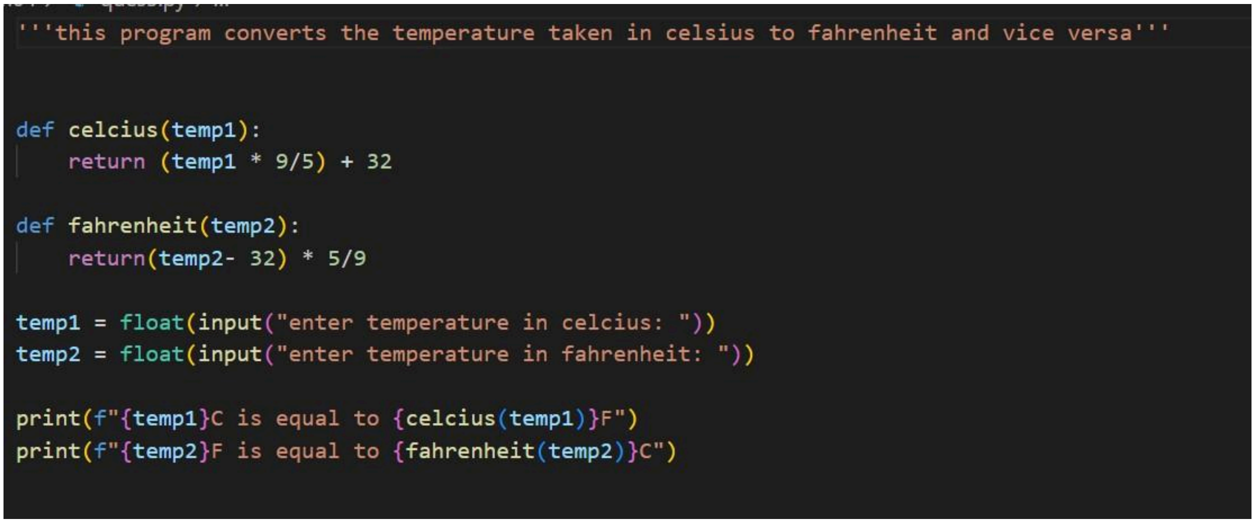


```python
'''this program always print the first letter of name in uppercase'''

name = input("hello, what is your name? ")

name2 = name.strip().capitalize()

print(f"hello, {name2}. nice to meet you!")
```

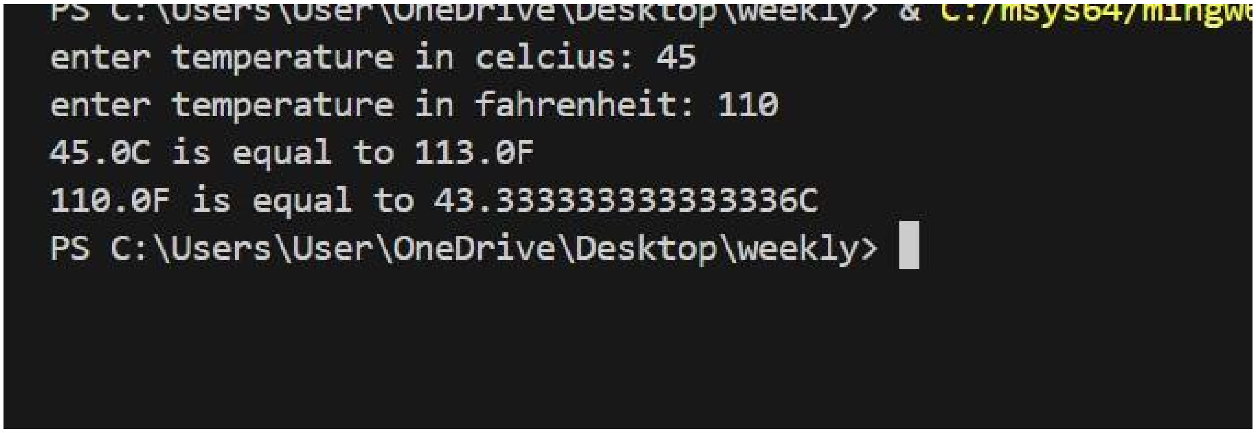


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/ms
hello, what is your name? rach
hello, Rach. nice to meet you!
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 5:



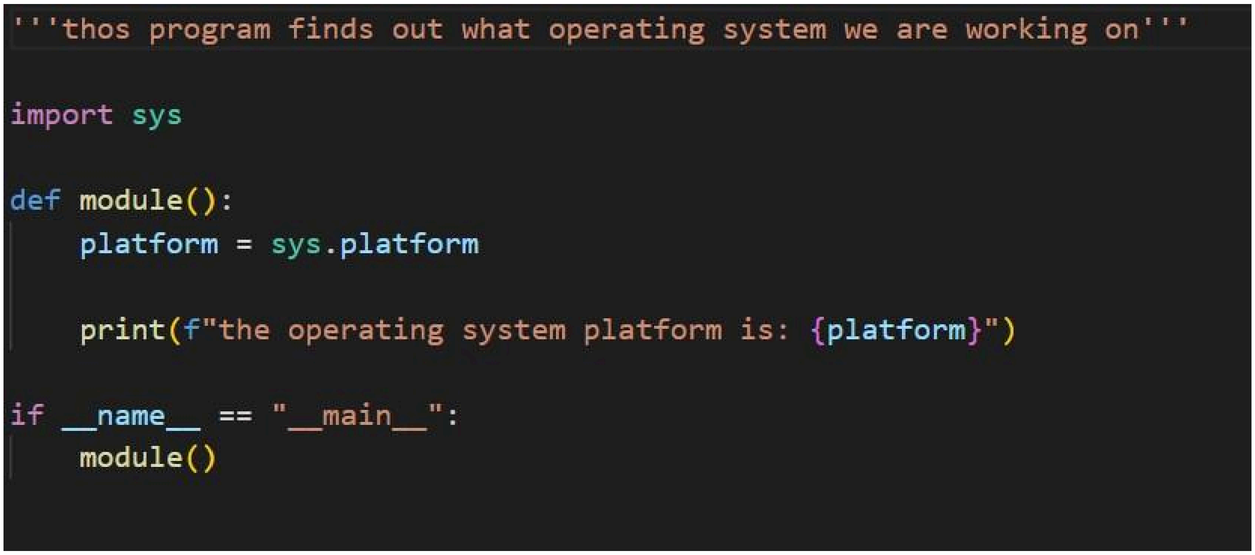
```
'''this program converts the temperature taken in celsius to fahrenheit and vice versa'''


def celcius(temp1):
    return (temp1 * 9/5) + 32

def fahrenheit(temp2):
    return(temp2- 32) * 5/9

temp1 = float(input("enter temperature in celcius: "))
temp2 = float(input("enter temperature in fahrenheit: "))

print(f"{temp1}C is equal to {celcius(temp1)}F")
print(f"{temp2}F is equal to {fahrenheit(temp2)}C")
```



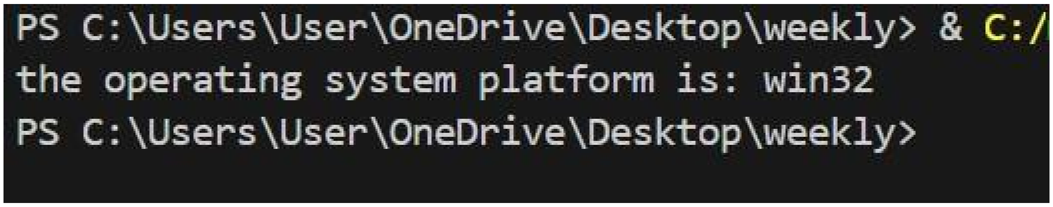
```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw
enter temperature in celcius: 45
enter temperature in fahrenheit: 110
45.0C is equal to 113.0F
110.0F is equal to 43.333333333333336C
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Week-5

Question 1:

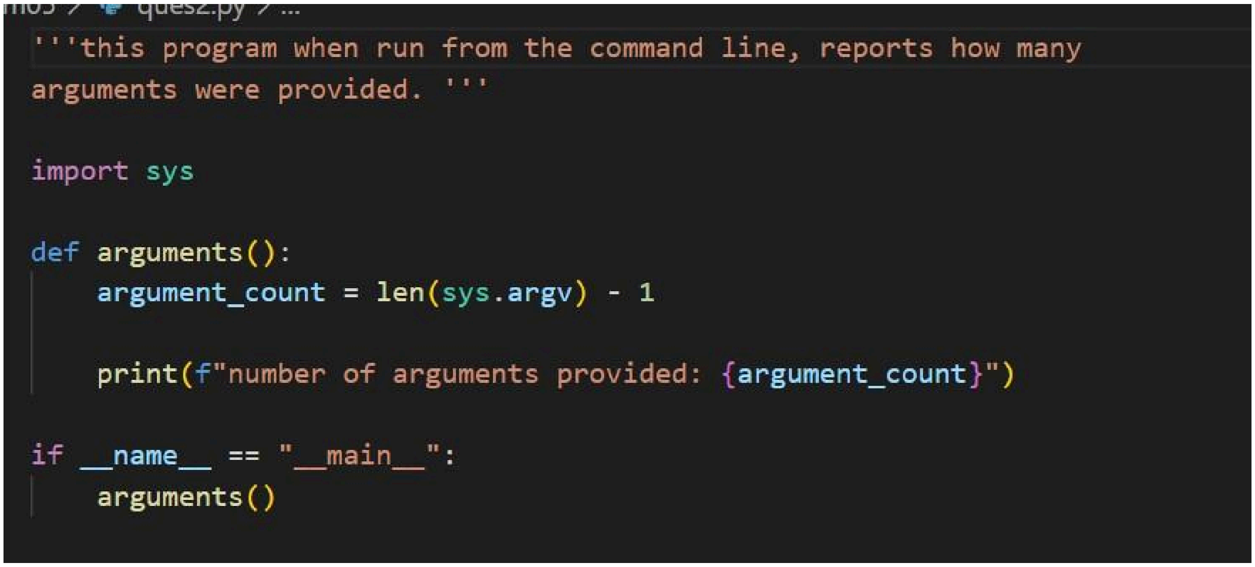


```python
'''thos program finds out what operating system we are working on'''

import sys

def module():
    platform = sys.platform

    print(f"the operating system platform is: {platform}")

if __name__ == "__main__":
    module()
```

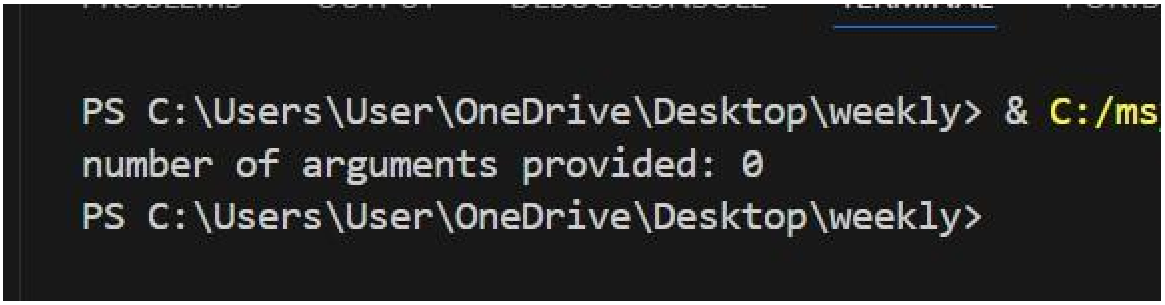


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/
the operating system platform is: win32
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 2:

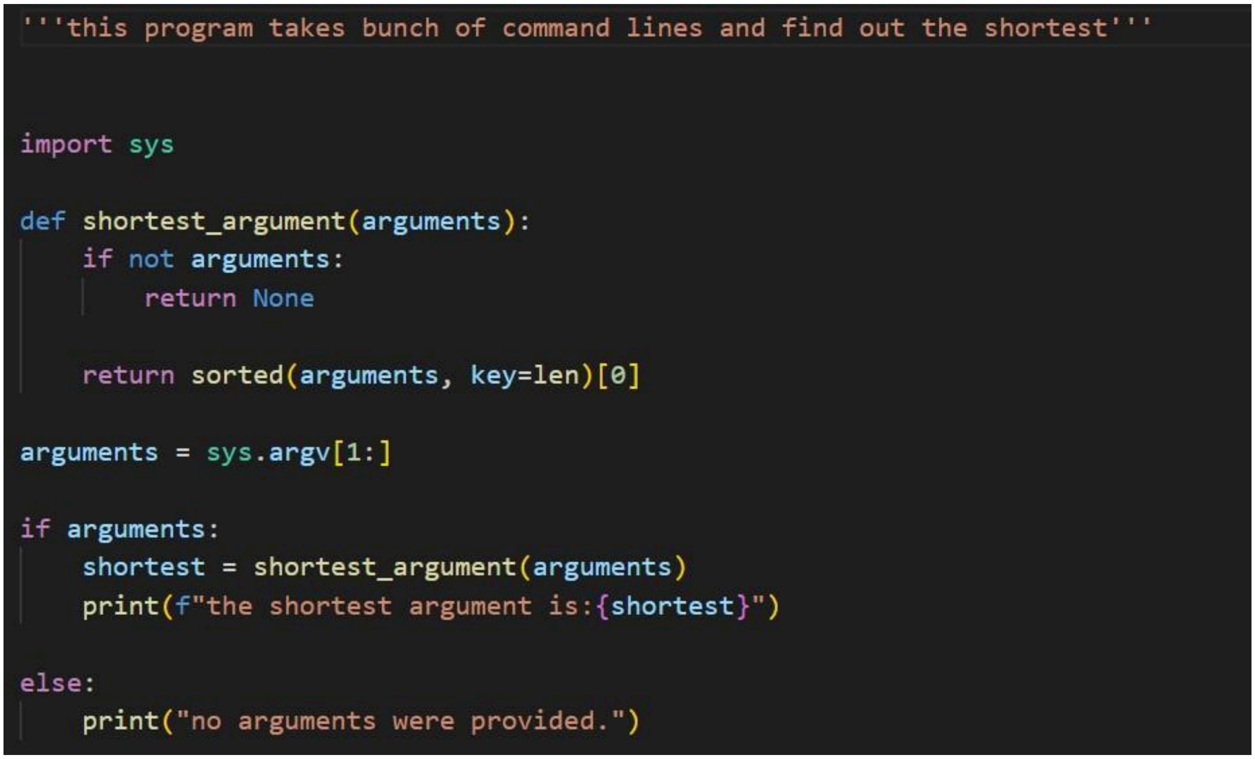


```python
'''this program when run from the command line, reports how many
arguments were provided. '''

import sys

def arguments():
    argument_count = len(sys.argv) - 1

    print(f"number of arguments provided: {argument_count}")

if __name__ == "__main__":
    arguments()
```

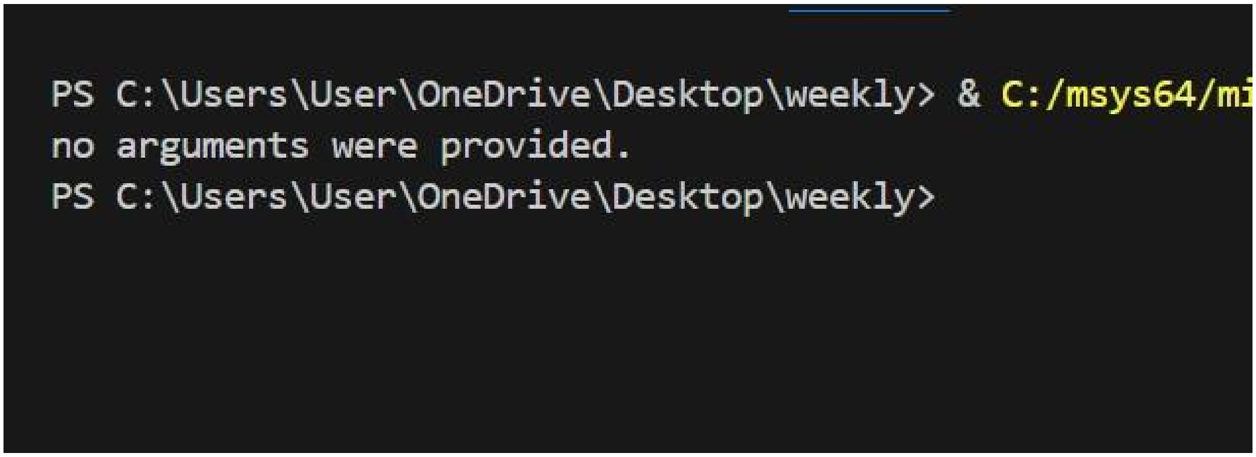


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/ms
number of arguments provided: 0
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 3:



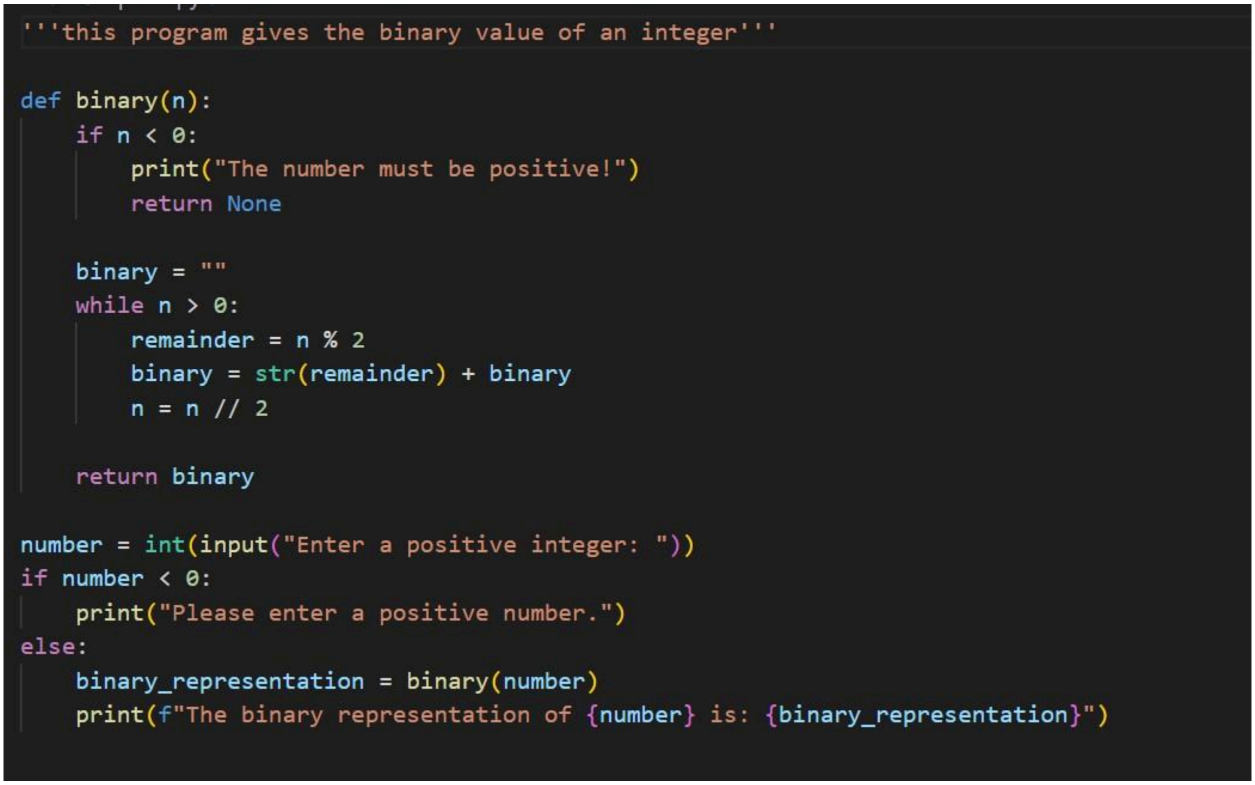
```
'''this program takes bunch of command lines and find out the shortest'''


import sys

def shortest_argument(arguments):
    if not arguments:
        return None

    return sorted(arguments, key=len)[0]

arguments = sys.argv[1:]

if arguments:
    shortest = shortest_argument(arguments)
    print(f"the shortest argument is:{shortest}")

else:
    print("no arguments were provided.")
```



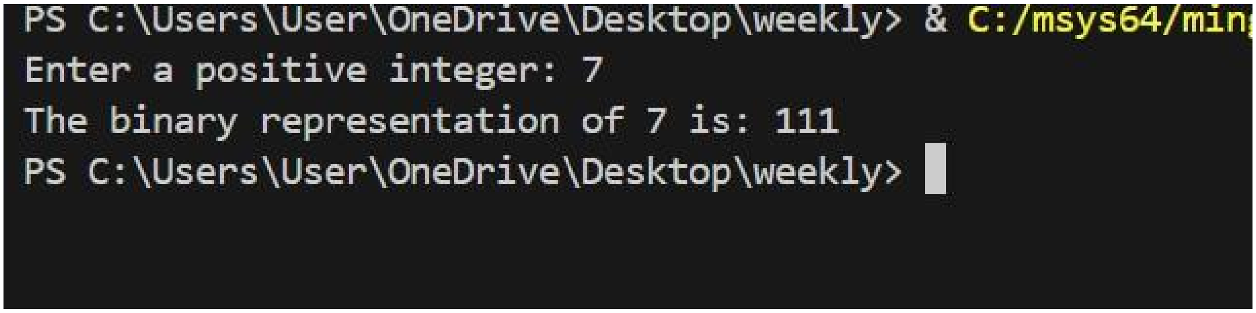
```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mi
no arguments were provided.
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Week-6

Question 1:

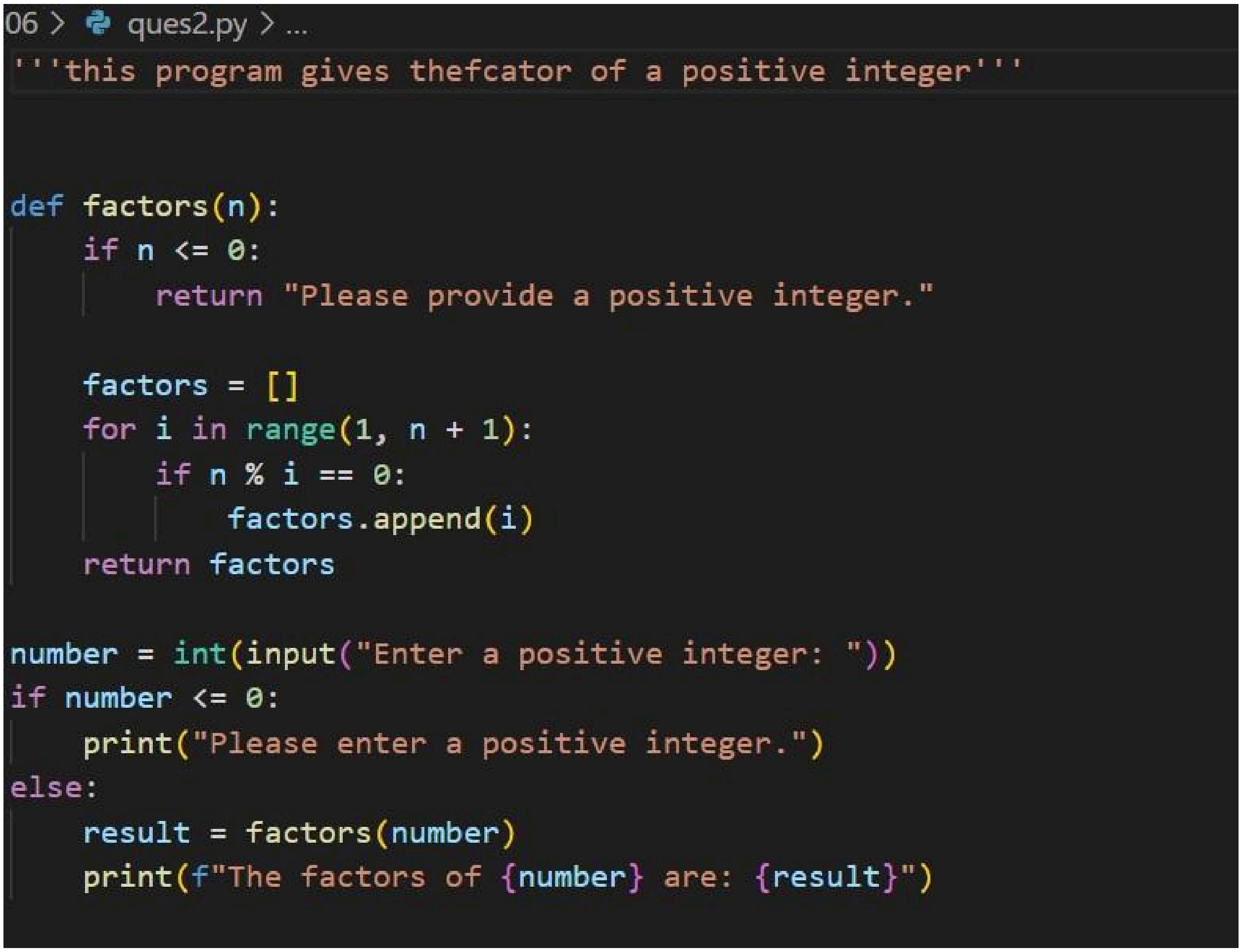


```python
'''this program gives the binary value of an integer'''

def binary(n):
    if n < 0:
        print("The number must be positive!")
        return None

    binary = ""
    while n > 0:
        remainder = n % 2
        binary = str(remainder) + binary
        n = n // 2

    return binary

number = int(input("Enter a positive integer: "))
if number < 0:
    print("Please enter a positive number.")
else:
    binary_representation = binary(number)
    print(f"The binary representation of {number} is: {binary_representation}")
```

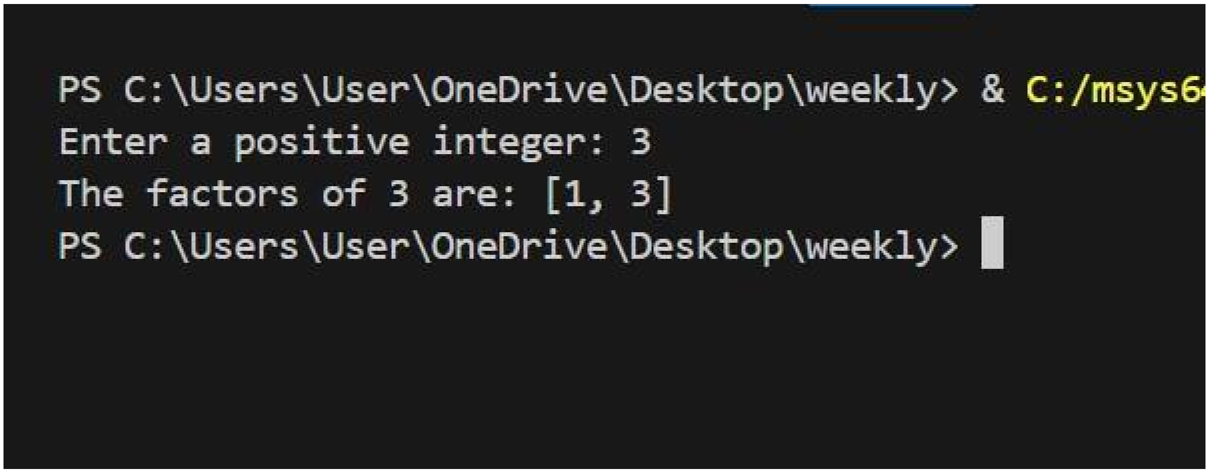


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/min
Enter a positive integer: 7
The binary representation of 7 is: 111
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 2:

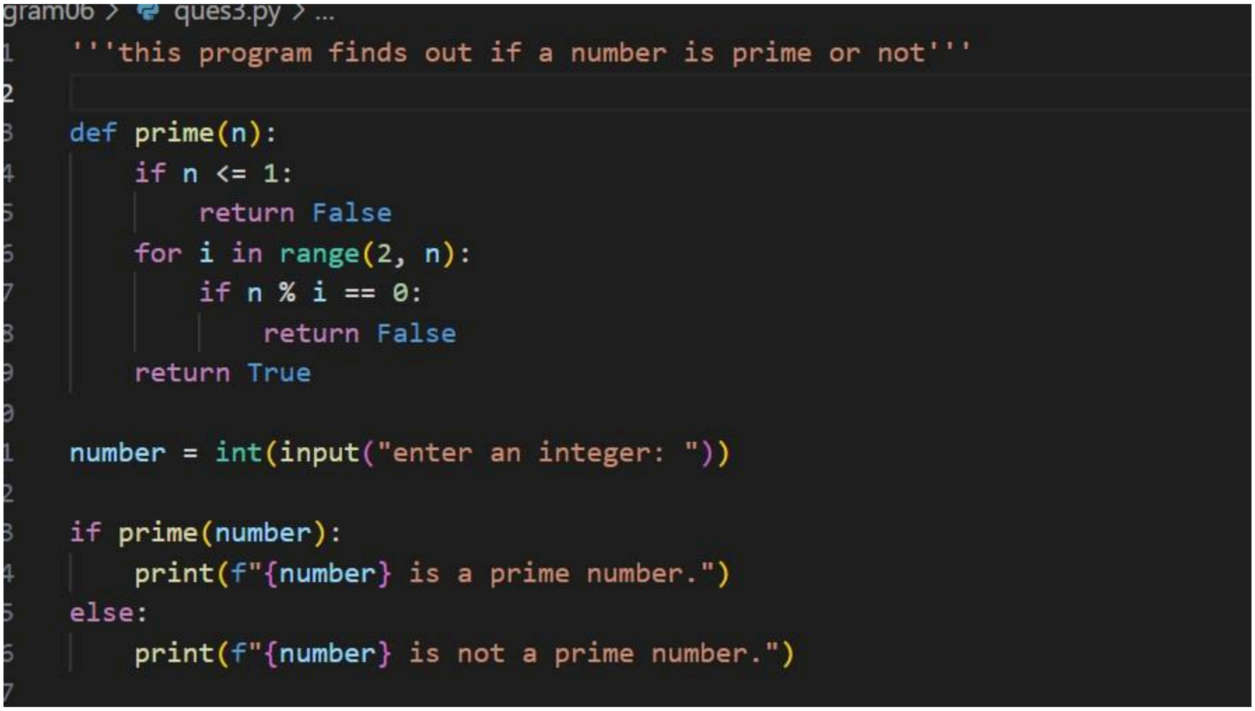


```
06 > ques2.py > ...
'''this program gives thefcator of a positive integer'''


def factors(n):
    if n <= 0:
        return "Please provide a positive integer."

    factors = []
    for i in range(1, n + 1):
        if n % i == 0:
            factors.append(i)
    return factors

number = int(input("Enter a positive integer: "))
if number <= 0:
    print("Please enter a positive integer.")
else:
    result = factors(number)
    print(f"The factors of {number} are: {result}")
```

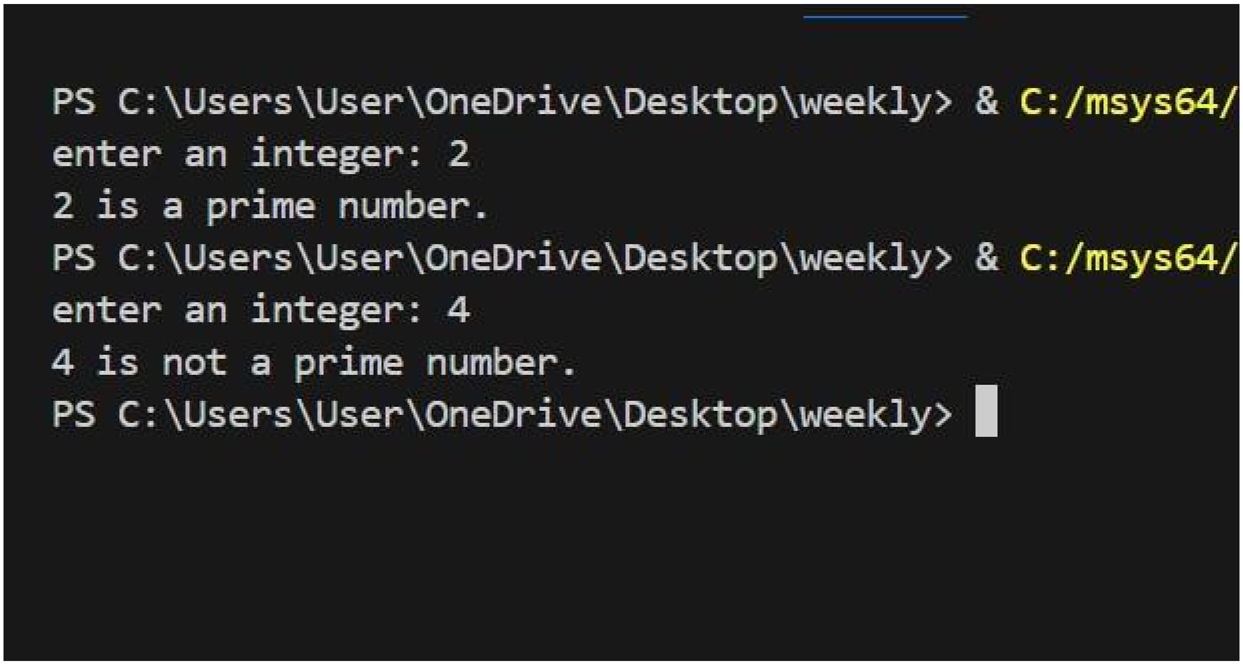


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys6
Enter a positive integer: 3
The factors of 3 are: [1, 3]
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 3:

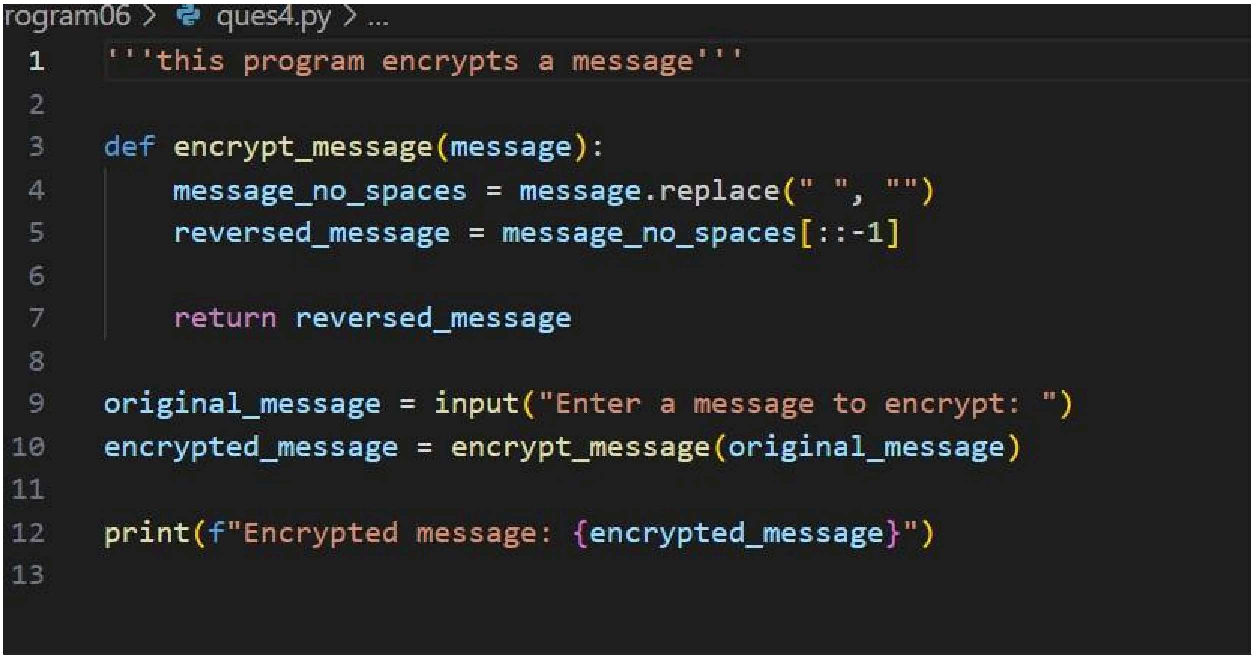


```python
'''this program finds out if a number is prime or not'''

def prime(n):
    if n <= 1:
        return False
    for i in range(2, n):
        if n % i == 0:
            return False
    return True

number = int(input("enter an integer: "))

if prime(number):
    print(f"{number} is a prime number.")
else:
    print(f"{number} is not a prime number.")
```

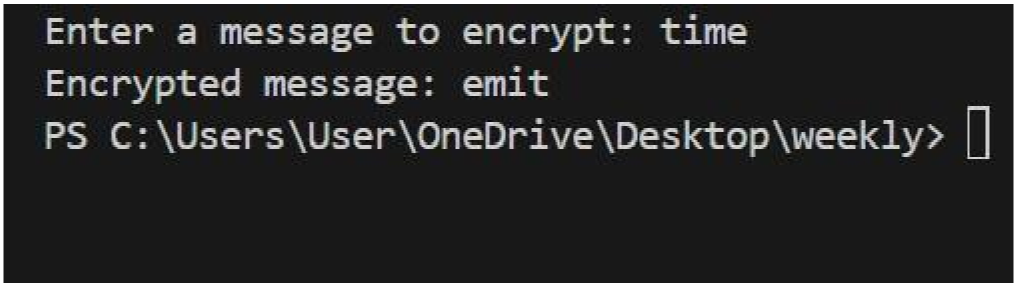


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/
enter an integer: 2
2 is a prime number.
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/
enter an integer: 4
4 is not a prime number.
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 4:

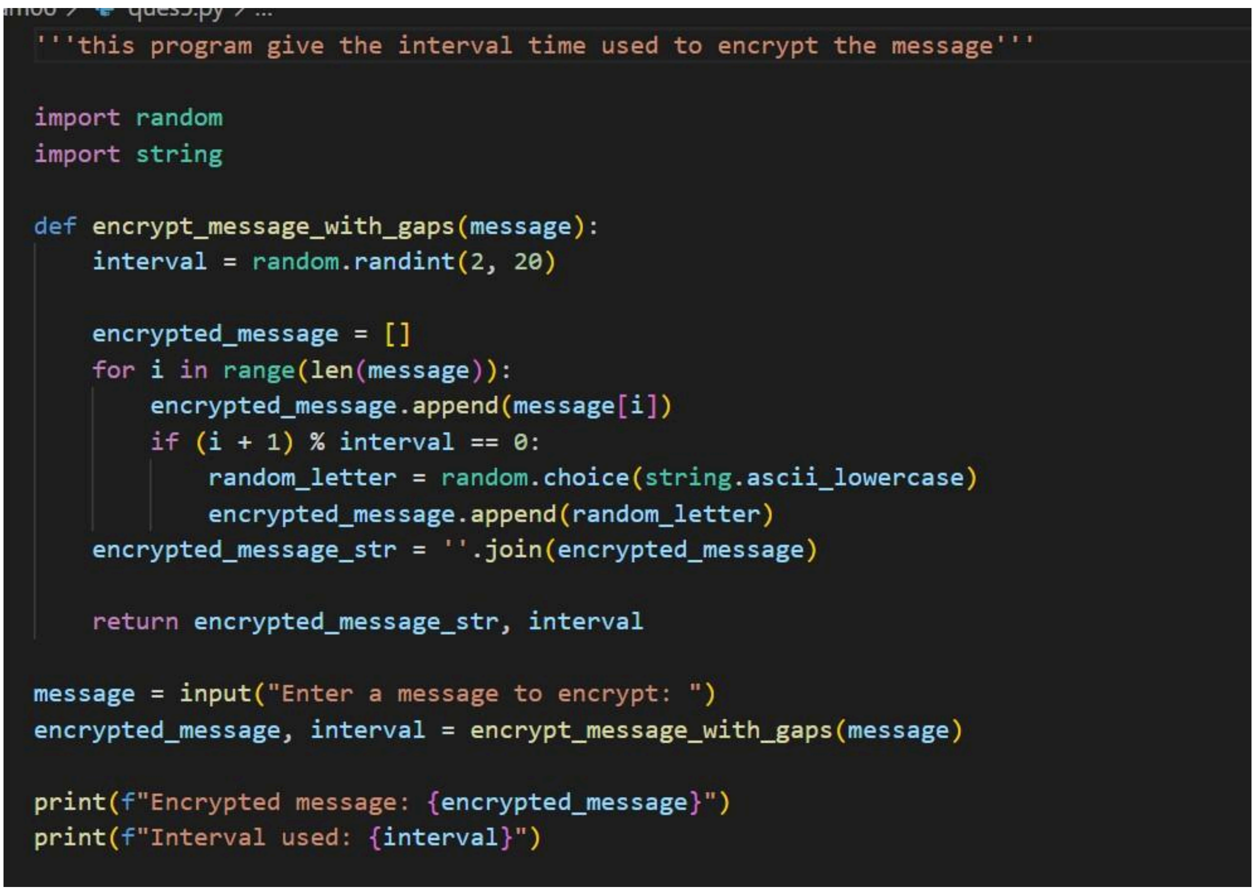


```python
'''this program encrypts a message'''

def encrypt_message(message):
    message_no_spaces = message.replace(" ", "")
    reversed_message = message_no_spaces[::-1]

    return reversed_message

original_message = input("Enter a message to encrypt: ")
encrypted_message = encrypt_message(original_message)

print(f"Encrypted message: {encrypted_message}")

```

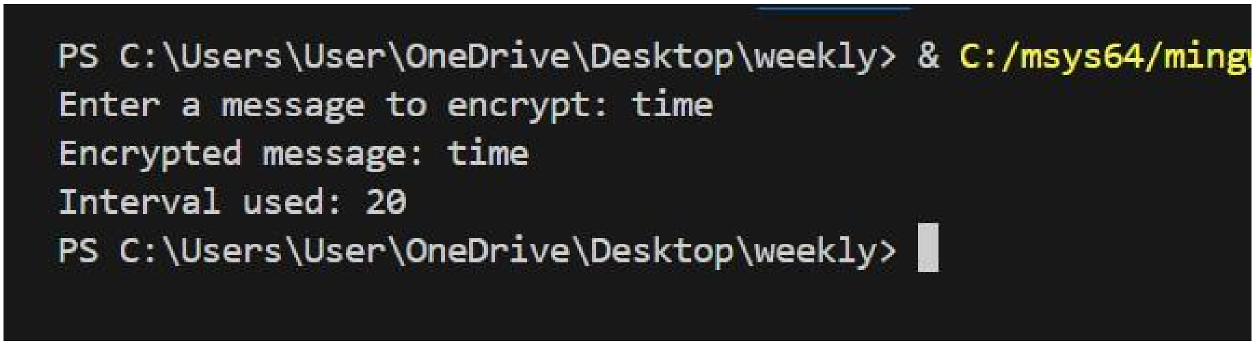


```
Enter a message to encrypt: time
Encrypted message: emit
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Question 5:

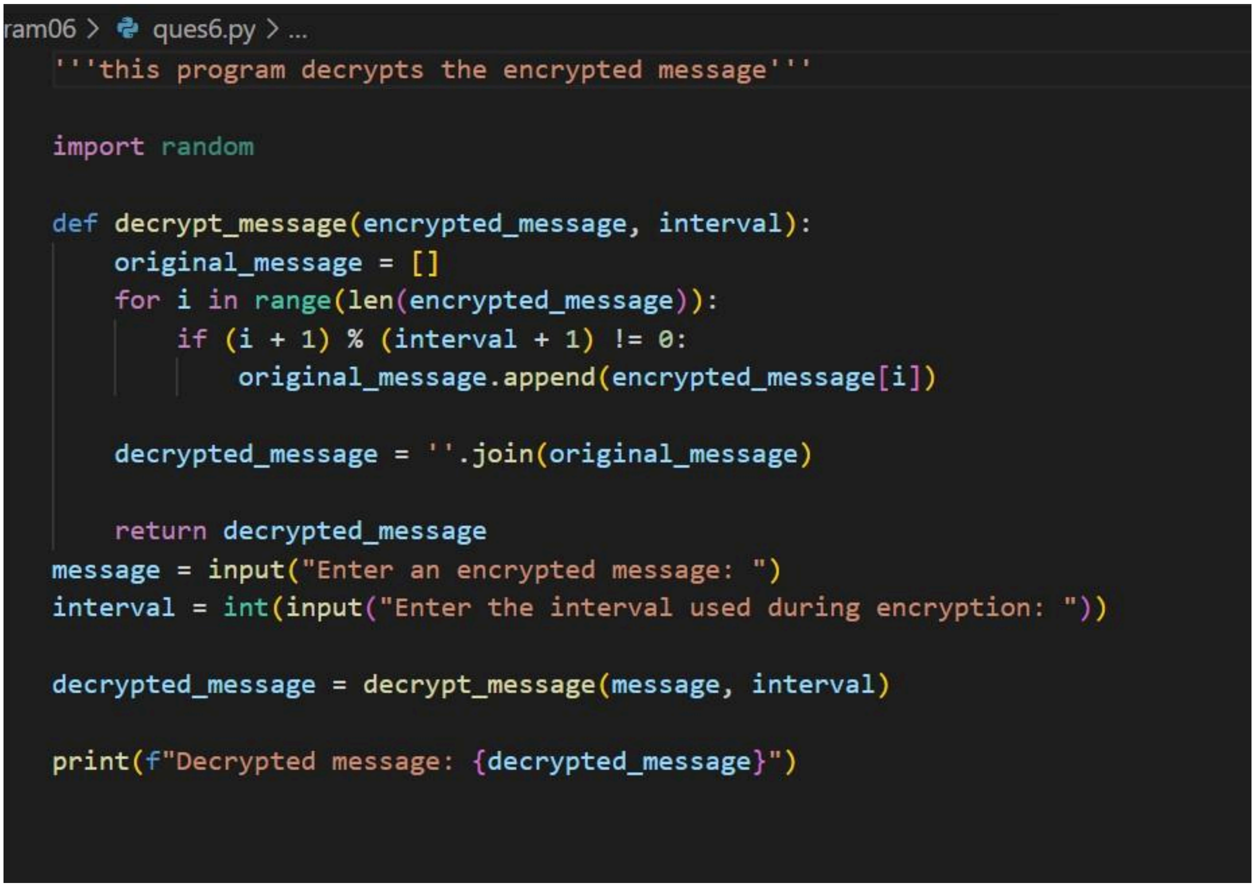


```python
'''this program give the interval time used to encrypt the message'''

import random
import string

def encrypt_message_with_gaps(message):
    interval = random.randint(2, 20)

    encrypted_message = []
    for i in range(len(message)):
        encrypted_message.append(message[i])
        if (i + 1) % interval == 0:
            random_letter = random.choice(string.ascii_lowercase)
            encrypted_message.append(random_letter)
    encrypted_message_str = ''.join(encrypted_message)

    return encrypted_message_str, interval

message = input("Enter a message to encrypt: ")
encrypted_message, interval = encrypt_message_with_gaps(message)

print(f"Encrypted message: {encrypted_message}")
print(f"Interval used: {interval}")
```

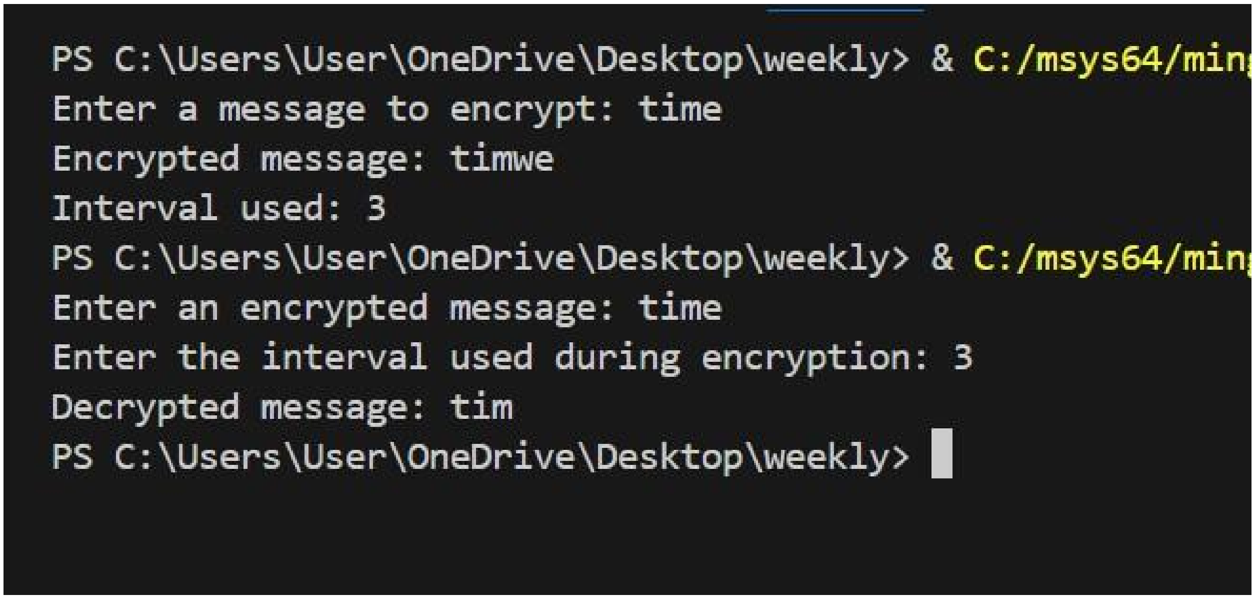


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/ming
Enter a message to encrypt: time
Encrypted message: time
Interval used: 20
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 6:



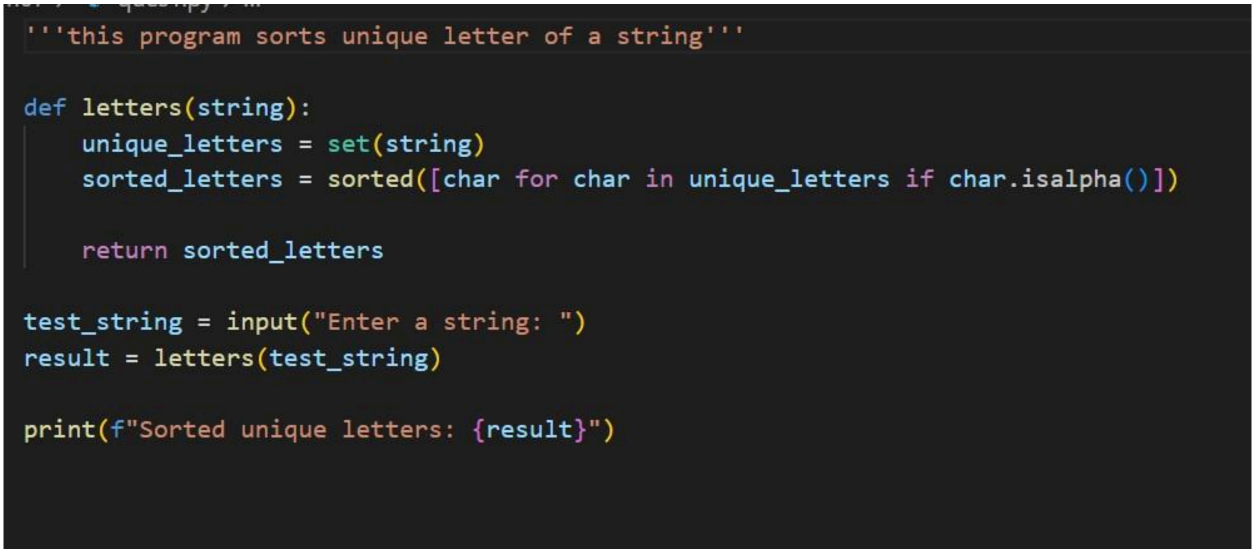
```python
'''this program decrypts the encrypted message'''

import random

def decrypt_message(encrypted_message, interval):
    original_message = []
    for i in range(len(encrypted_message)):
        if (i + 1) % (interval + 1) != 0:
            original_message.append(encrypted_message[i])

    decrypted_message = ''.join(original_message)

    return decrypted_message
message = input("Enter an encrypted message: ")
interval = int(input("Enter the interval used during encryption: "))

decrypted_message = decrypt_message(message, interval)

print(f"Decrypted message: {decrypted_message}")
```



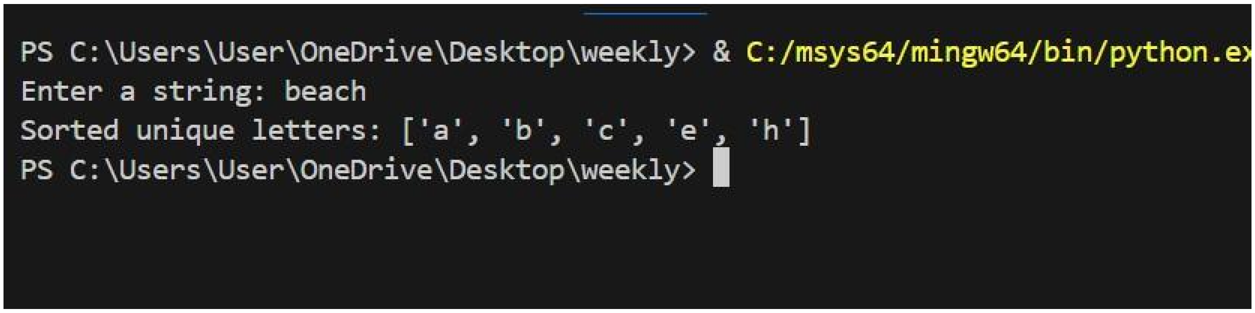
```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/min
Enter a message to encrypt: time
Encrypted message: timwe
Interval used: 3
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/min
Enter an encrypted message: time
Enter the interval used during encryption: 3
Decrypted message: tim
PS C:\Users\User\OneDrive\Desktop\weekly> 
```

Week-7

Question 1:

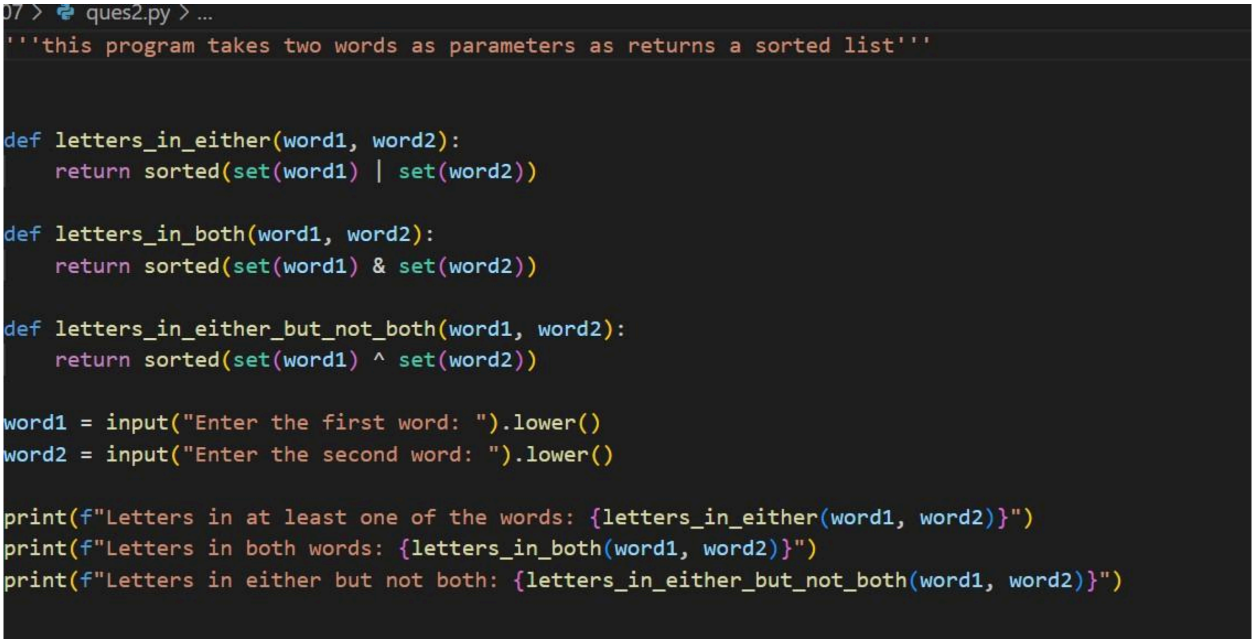


```python
'''this program sorts unique letter of a string'''

def letters(string):
    unique_letters = set(string)
    sorted_letters = sorted([char for char in unique_letters if char.isalpha()])

    return sorted_letters

test_string = input("Enter a string: ")
result = letters(test_string)

print(f"Sorted unique letters: {result}")
```

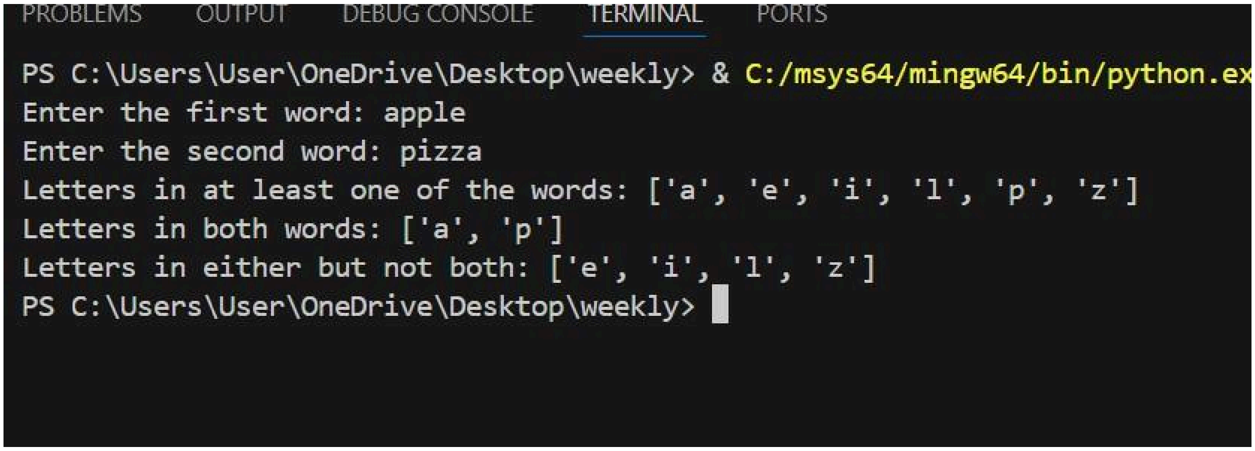


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/python.ex
Enter a string: beach
Sorted unique letters: ['a', 'b', 'c', 'e', 'h']
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 2:

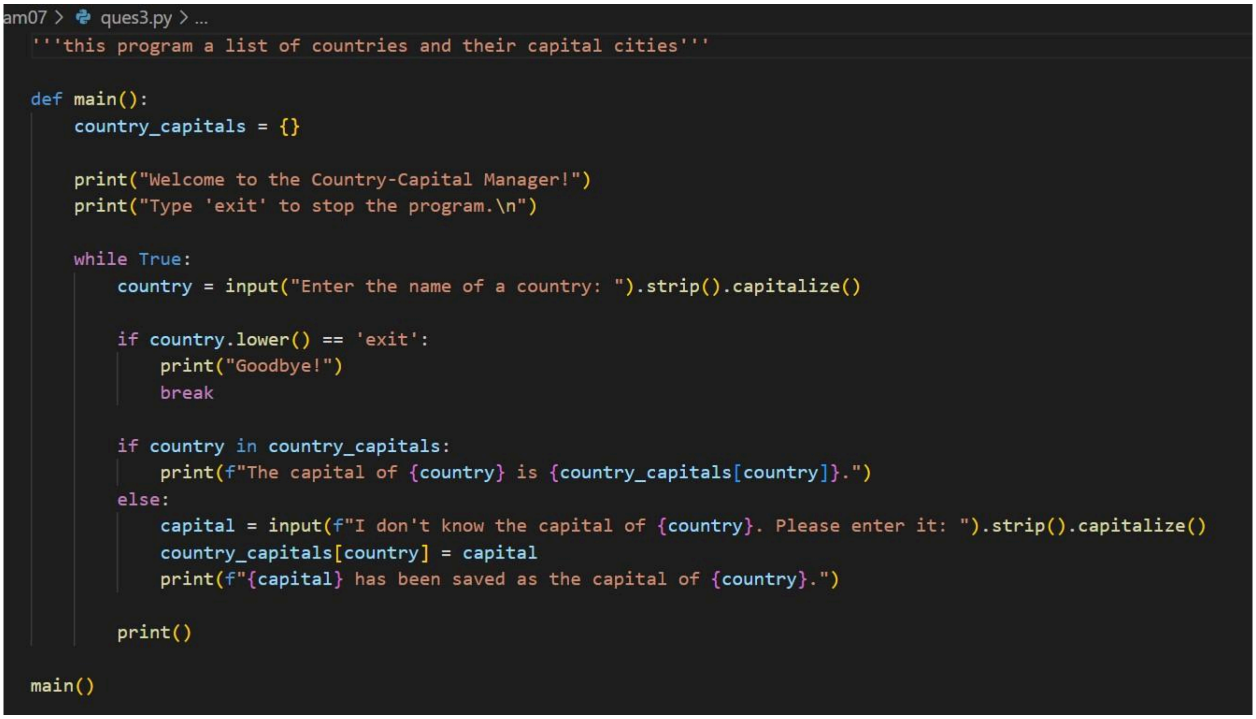


```
'''this program takes two words as parameters as returns a sorted list'''


def letters_in_either(word1, word2):
    return sorted(set(word1) | set(word2))

def letters_in_both(word1, word2):
    return sorted(set(word1) & set(word2))

def letters_in_either_but_not_both(word1, word2):
    return sorted(set(word1) ^ set(word2))

word1 = input("Enter the first word: ").lower()
word2 = input("Enter the second word: ").lower()

print(f"Letters in at least one of the words: {letters_in_either(word1, word2)}")
print(f"Letters in both words: {letters_in_both(word1, word2)}")
print(f"Letters in either but not both: {letters_in_either_but_not_both(word1, word2)}")
```

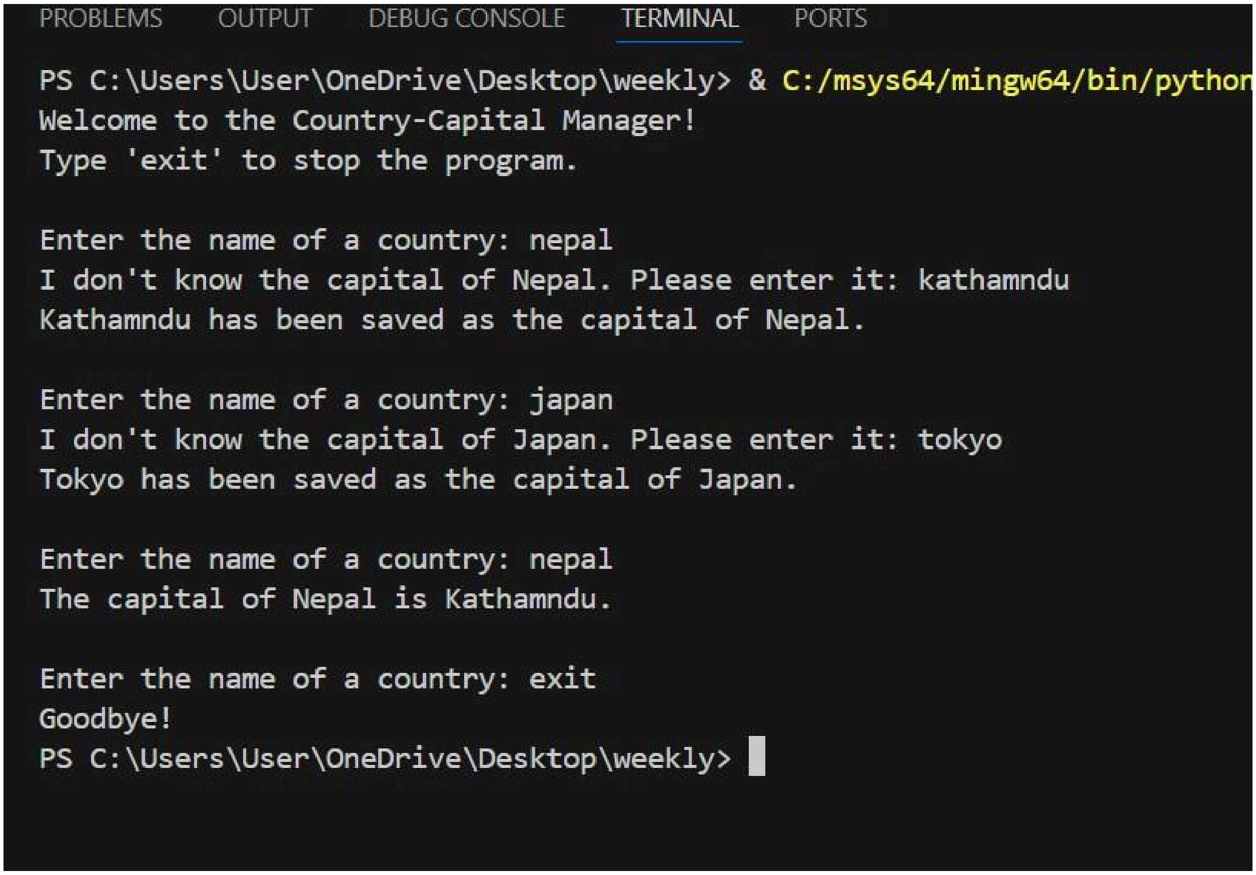


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/python.ex
Enter the first word: apple
Enter the second word: pizza
Letters in at least one of the words: ['a', 'e', 'i', 'l', 'p', 'z']
Letters in both words: ['a', 'p']
Letters in either but not both: ['e', 'i', 'l', 'z']
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 3:

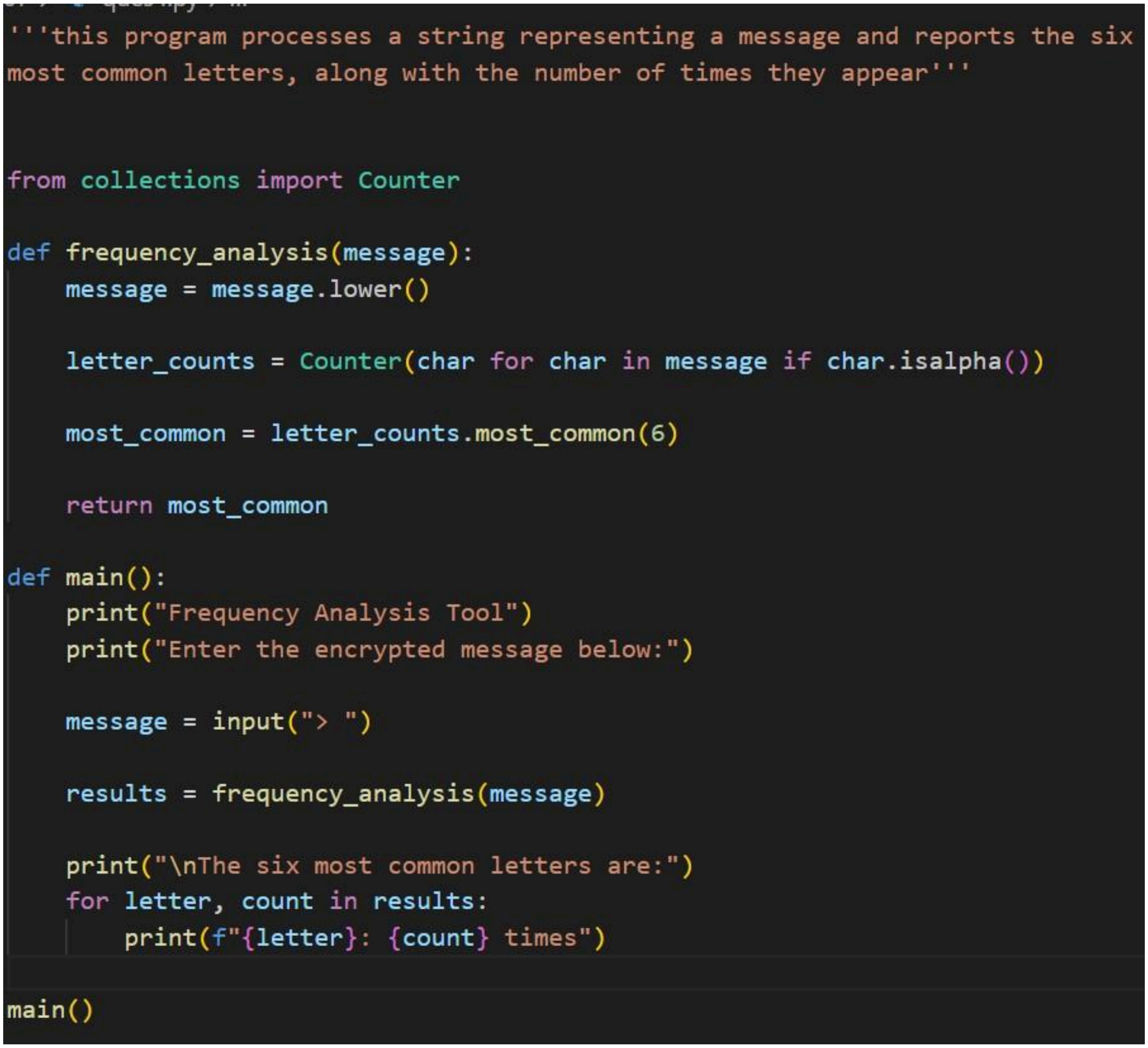


```python
'''this program a list of countries and their capital cities'''

def main():
    country_capitals = {}

    print("Welcome to the Country-Capital Manager!")
    print("Type 'exit' to stop the program.\n")

    while True:
        country = input("Enter the name of a country: ").strip().capitalize()

        if country.lower() == 'exit':
            print("Goodbye!")
            break

        if country in country_capitals:
            print(f"The capital of {country} is {country_capitals[country]}.")
        else:
            capital = input(f"I don't know the capital of {country}. Please enter it: ").strip().capitalize()
            country_capitals[country] = capital
            print(f"{capital} has been saved as the capital of {country}.")

        print()

main()
```

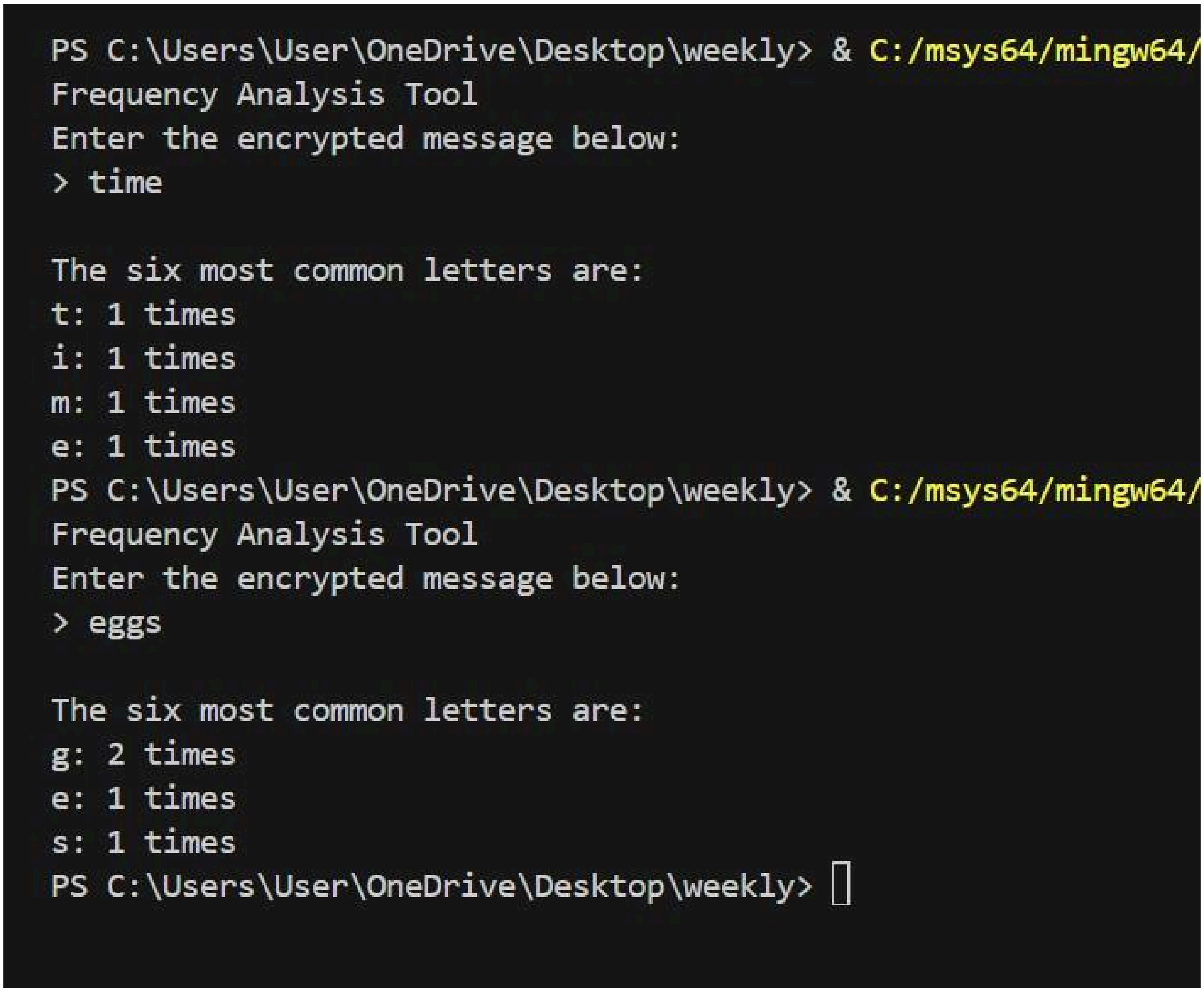


```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/bin/python
Welcome to the Country-Capital Manager!
Type 'exit' to stop the program.

Enter the name of a country: nepal
I don't know the capital of Nepal. Please enter it: kathamndu
Kathamndu has been saved as the capital of Nepal.

Enter the name of a country: japan
I don't know the capital of Japan. Please enter it: tokyo
Tokyo has been saved as the capital of Japan.

Enter the name of a country: nepal
The capital of Nepal is Kathamndu.

Enter the name of a country: exit
Goodbye!
PS C:\Users\User\OneDrive\Desktop\weekly>
```

Question 4:



```
'''this program processes a string representing a message and reports the six
most common letters, along with the number of times they appear'''


from collections import Counter

def frequency_analysis(message):
    message = message.lower()

    letter_counts = Counter(char for char in message if char.isalpha())

    most_common = letter_counts.most_common(6)

    return most_common

def main():
    print("Frequency Analysis Tool")
    print("Enter the encrypted message below:")

    message = input("> ")

    results = frequency_analysis(message)

    print("\nThe six most common letters are:")
    for letter, count in results:
        print(f"{letter}: {count} times")


main()
```



```
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/
Frequency Analysis Tool
Enter the encrypted message below:
> time

The six most common letters are:
t: 1 times
i: 1 times
m: 1 times
e: 1 times
PS C:\Users\User\OneDrive\Desktop\weekly> & C:/msys64/mingw64/
Frequency Analysis Tool
Enter the encrypted message below:
> eggs

The six most common letters are:
g: 2 times
e: 1 times
s: 1 times
PS C:\Users\User\OneDrive\Desktop\weekly> 
```