Name: ALAN GEORGE MATHEWS
KTU-ID: MDL20CS016

Assignment 1: Programming in Python CST 362

- **Date of submission:27-Feb-2023 before 10am**
- **Learning outcome: Python basics, operators, modules/packages, control statements**

1. **Python is developed by…….How Python got its name**

Ans: Python was developed by Guido van Rossum. Monty Python’s Flying Circus was a very famous comedy BBC TV series from the 70s. It was rated so well that it was like a must-watch series at the time and also was known to be very unpredictive, creative, and random; basically, it talked about everything. The python programming language was very famous since it is easy to learn and use and was considered to be the must-know programming language. Guido van Rossum wanted to give his programming language a name that was unique, mysterious, and short, and thus, Python was named after Monty Python’s Flying Circus. 2. Write the output

2. **Write the output of the statement 2**3**2**

Ans: 512

3. **What is the difference between / and // operators. Write one example**

Ans: The / operator performs regular division, which means it returns a floating-point number that represents the quotient of the division operation. The // operator performs integer division, which means it returns the quotient of the division operation as an integer by discarding any fractional part. Eg: Input: 10/4 -> Output: 2.5 —--- Input: 10//4 -> Output: 2

4. **How the expression 2+3*2**2/3 is evaluated show the order of evaluation and final output**

Ans: The expression will be evaluated as follows: 2 + ((3 * (2 ** 2)) / 3) 2 + ((3 * 4) / 3) 2 + (12 / 3) 2 + 4 6 Final output will be 6.

5. **How single line and multiline comments are added in Python script**

Ans: In Python, we use the hash symbol (#) to add single-line comments and triple quotes (""" """ or ''' ''') to add multi-line comments. Single-line comments are used to add a brief description or explanation of code on the same line. Multi-line comments are often used to provide a description of a module, function, or class in Python

6. **Write the logical operators**

Ans: In Python, the logical operators are AND, OR, and NOT. These operators are used to perform logical operations on Boolean values. and: Returns True if both operands are True, otherwise returns False. or: Returns True if at least one operand is True, otherwise returns False. not: Returns the opposite of the operand's value. If the operand is True, it returns False, and if the operand is False, it returns True.

7. **What are bit wise operators**

Ans: Bitwise operators perform operations on the binary representation of the operands at the bit level. The bitwise operators are: &: Bitwise AND operator |: Bitwise OR operator ^: Bitwise XOR operator ~: Bitwise NOT operator <<: Left shift operator >>: Right shift operator

8. **x=0xAA, y=0o16, z=0b10110 , find x^y|z**

Ans. 182

9. **How to get the last bit of a number. Write the bitwise operation**
   e/g: x=2 o/p:1 x=3 o/p:1

Ans.
Num =3
last_bit = num & 1

10. x=12, what is the output of x<<2 justify your answer

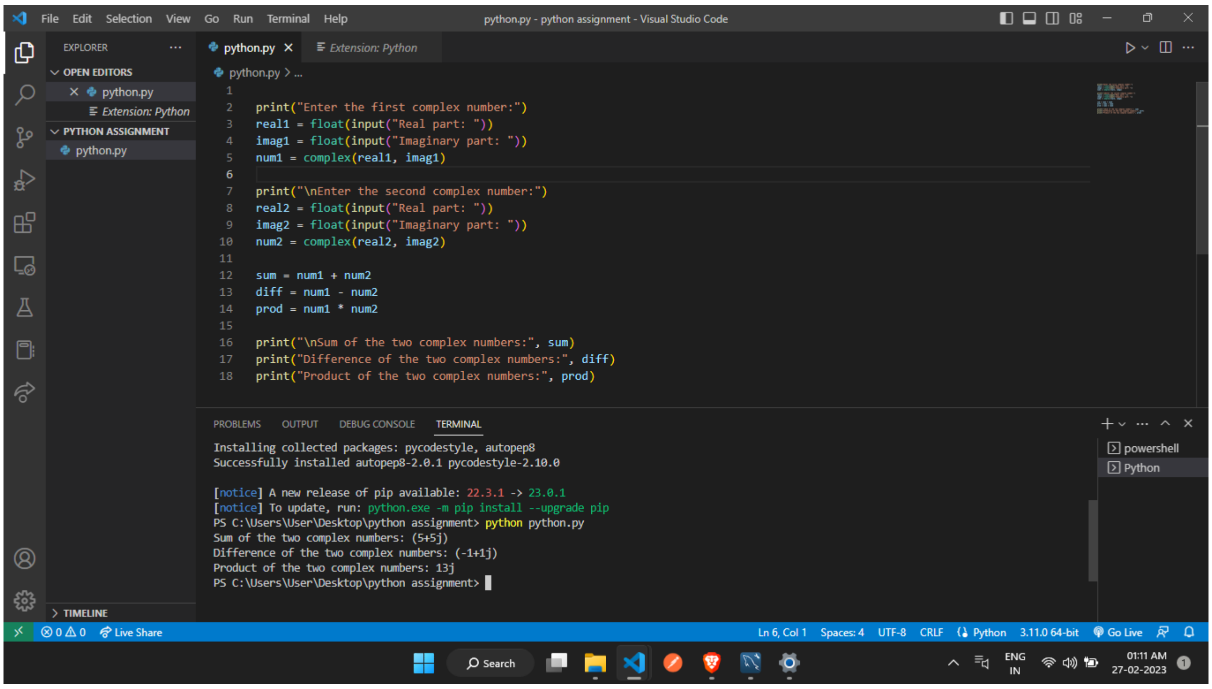
Ans: In Python, the << operator is the left shift operator, which shifts the bits of a number to the left by a certain number of positions. Each shift to the left doubles the value of the number. Thus, result for this problem would mean x becomes 48 x in binary: 00001100(12)
Result: 00110000(48)

11. What are Boolean data types

Ans: In Python, the Boolean data type is a built-in data type that represents the two truth values True and False. These values are used to represent the logical values of True and False in Python programs. Boolean values are often used in conditional statements and loops to control program flow.

- ## Basic Scripting

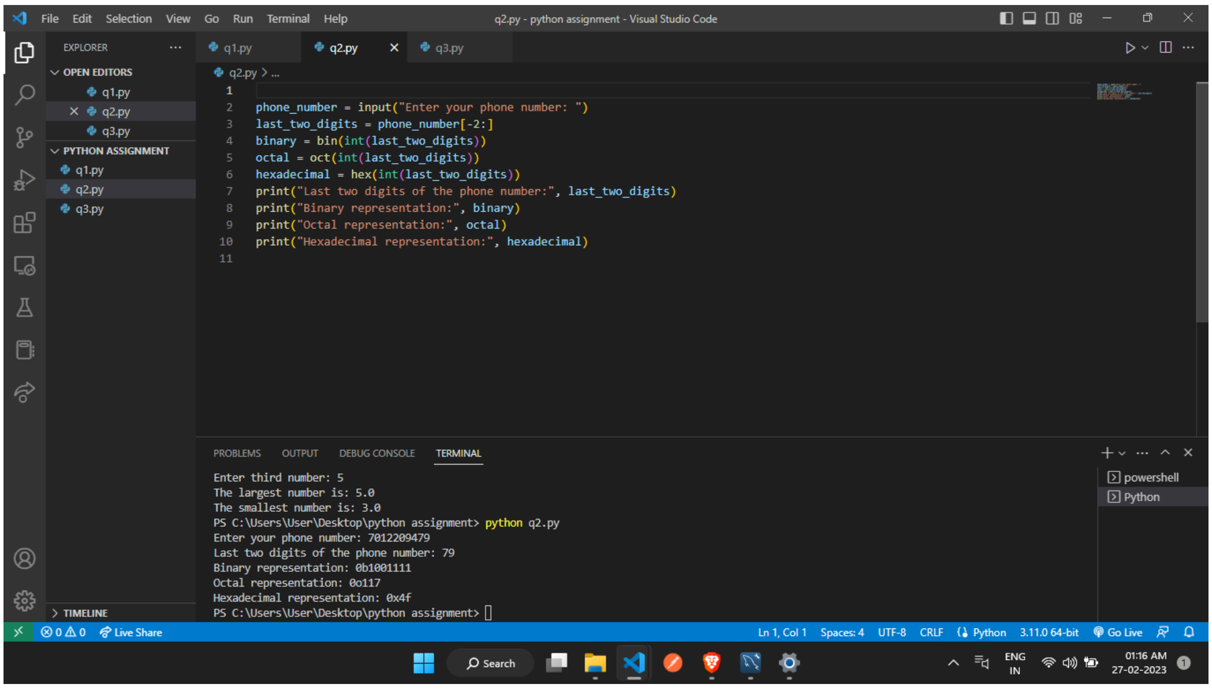
1. **Write a Python script which will read two complex numbers and find their sum, difference and product.**



```python

print("Enter the first complex number:")
real1 = float(input("Real part: "))
imag1 = float(input("Imaginary part: "))
num1 = complex(real1, imag1)

print("\nEnter the second complex number:")
real2 = float(input("Real part: "))
imag2 = float(input("Imaginary part: "))
num2 = complex(real2, imag2)

sum = num1 + num2
diff = num1 - num2
prod = num1 * num2

print("\nSum of the two complex numbers:", sum)
print("Difference of the two complex numbers:", diff)
print("Product of the two complex numbers:", prod)
```

```
Installing collected packages: pycodestyle, autopep8
Successfully installed autopep8-2.0.1 pycodestyle-2.10.0

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\User\Desktop\python assignment> python python.py
Sum of the two complex numbers: (5+5j)
Difference of the two complex numbers: (-1+1j)
Product of the two complex numbers: 13j
PS C:\Users\User\Desktop\python assignment>
```

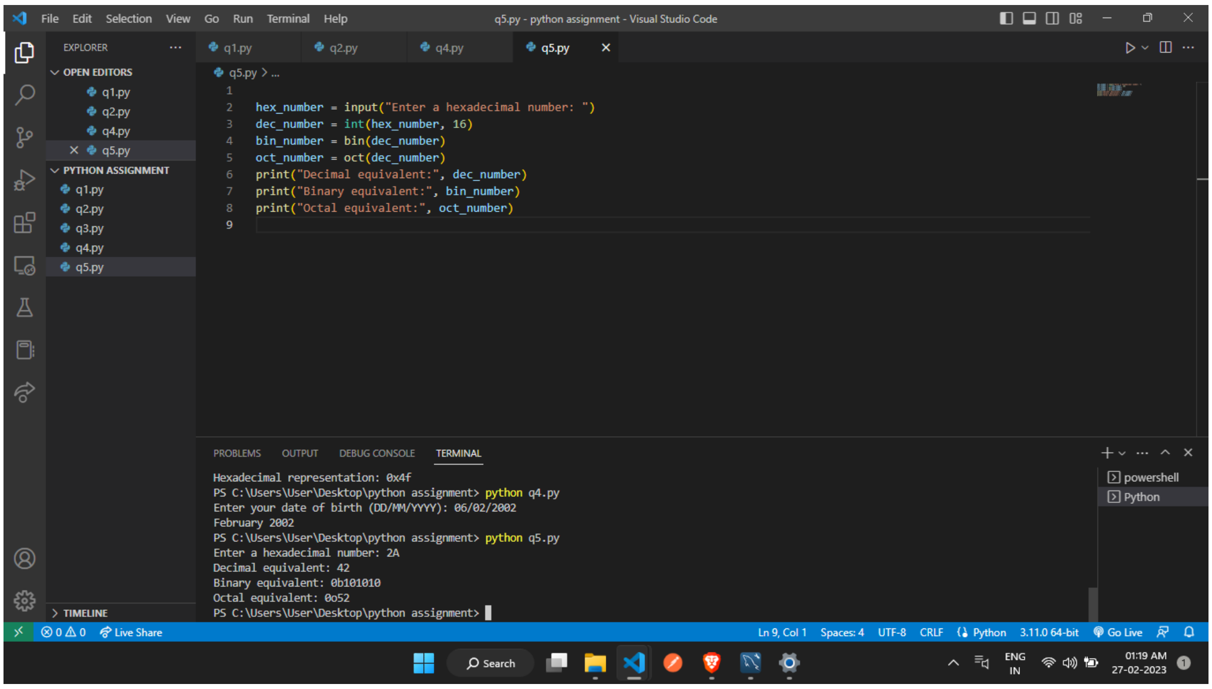
2. Read your phone number and print the last two digit in binary, octal and hex.



```python

phone_number = input("Enter your phone number: ")
last_two_digits = phone_number[-2:]
binary = bin(int(last_two_digits))
octal = oct(int(last_two_digits))
hexadecimal = hex(int(last_two_digits))
print("Last two digits of the phone number:", last_two_digits)
print("Binary representation:", binary)
print("Octal representation:", octal)
print("Hexadecimal representation:", hexadecimal)

```

```
Enter third number: 5
The largest number is: 5.0
The smallest number is: 3.0
PS C:\Users\User\Desktop\python assignment> python q2.py
Enter your phone number: 7012209479
Last two digits of the phone number: 79
Binary representation: 0b1001111
Octal representation: 0o117
Hexadecimal representation: 0x4f
PS C:\Users\User\Desktop\python assignment>
```

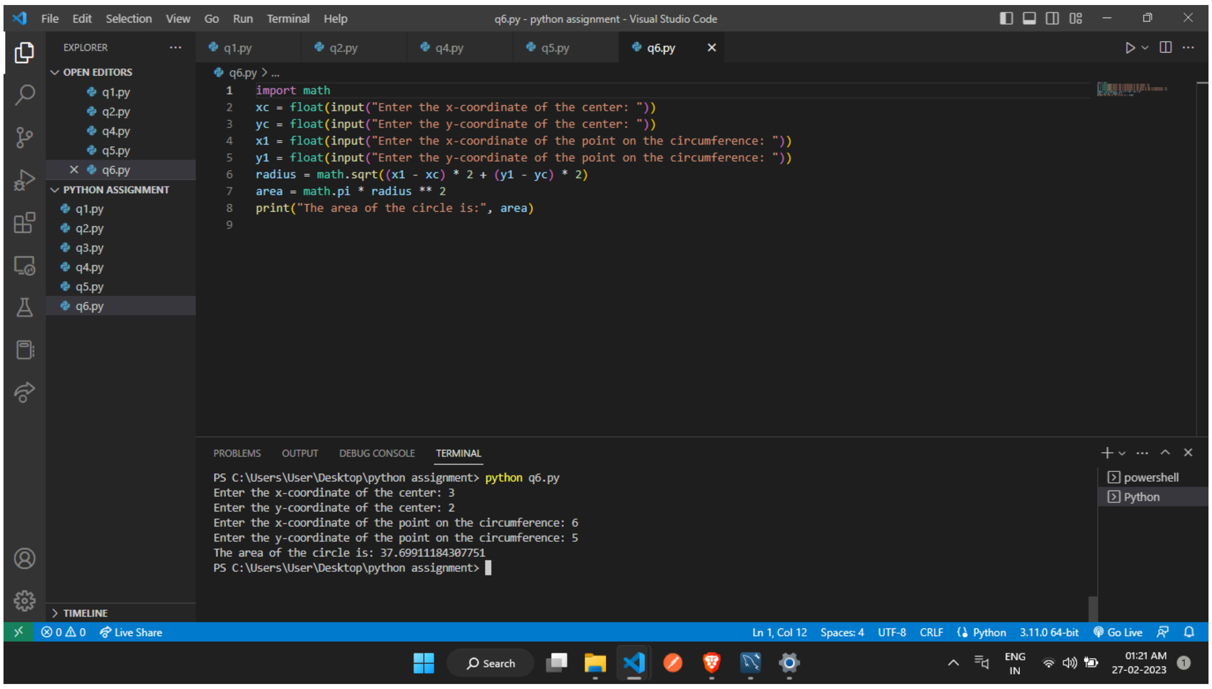
3. Read a hexa decimal number and print the dec, bin and octal equivalent.



```python

hex_number = input("Enter a hexadecimal number: ")
dec_number = int(hex_number, 16)
bin_number = bin(dec_number)
oct_number = oct(dec_number)
print("Decimal equivalent:", dec_number)
print("Binary equivalent:", bin_number)
print("Octal equivalent:", oct_number)

```

```
Hexadecimal representation: 0x4f
PS C:\Users\User\Desktop\python assignment> python q4.py
Enter your date of birth (DD/MM/YYYY): 06/02/2002
February 2002
PS C:\Users\User\Desktop\python assignment> python q5.py
Enter a hexadecimal number: 2A
Decimal equivalent: 42
Binary equivalent: 0b101010
Octal equivalent: 0o52
PS C:\Users\User\Desktop\python assignment>
```

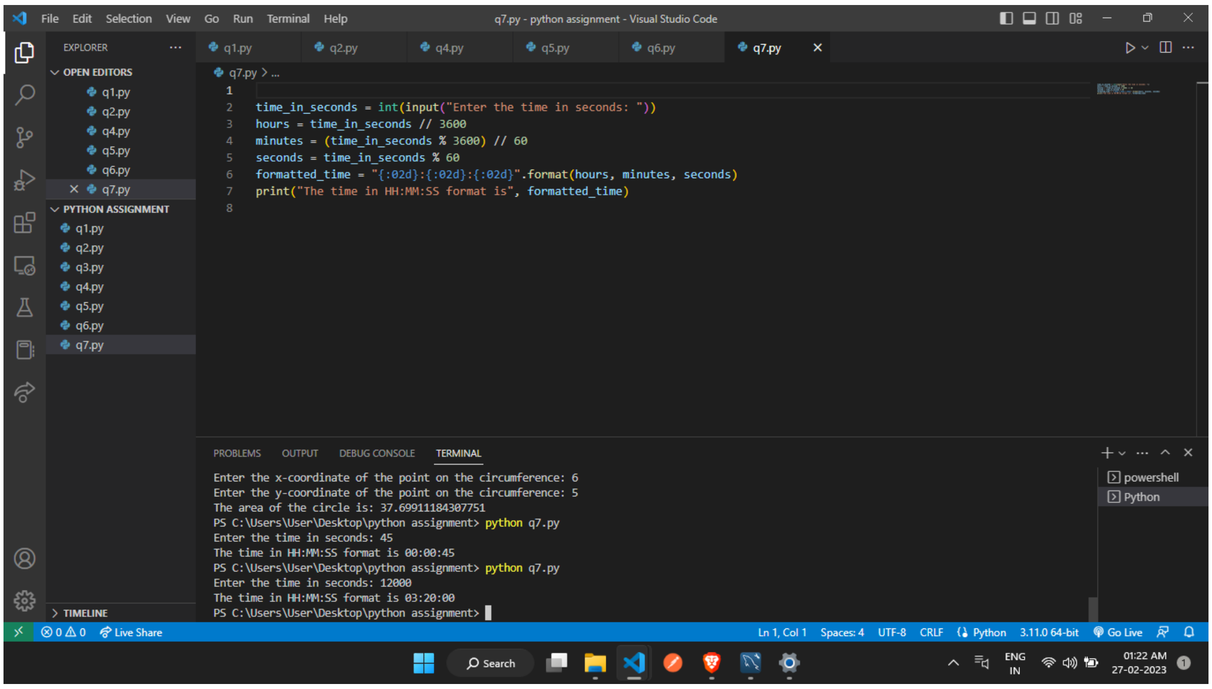
4. **Given the center (xc,yc) and a point on the circle(x1,y1). Find the area**



```python
import math
xc = float(input("Enter the x-coordinate of the center: "))
yc = float(input("Enter the y-coordinate of the center: "))
x1 = float(input("Enter the x-coordinate of the point on the circumference: "))
y1 = float(input("Enter the y-coordinate of the point on the circumference: "))
radius = math.sqrt((x1 - xc) * 2 + (y1 - yc) * 2)
area = math.pi * radius ** 2
print("The area of the circle is:", area)
```

```
PS C:\Users\User\Desktop\python assignment> python q6.py
Enter the x-coordinate of the center: 3
Enter the y-coordinate of the center: 2
Enter the x-coordinate of the point on the circumference: 6
Enter the y-coordinate of the point on the circumference: 5
The area of the circle is: 37.69911184307751
PS C:\Users\User\Desktop\python assignment>
```

5. **Write a Python program to read time in seconds and Print in HH:MM:SS format.**
i/p in seconds :1000
o/p:00:06:40

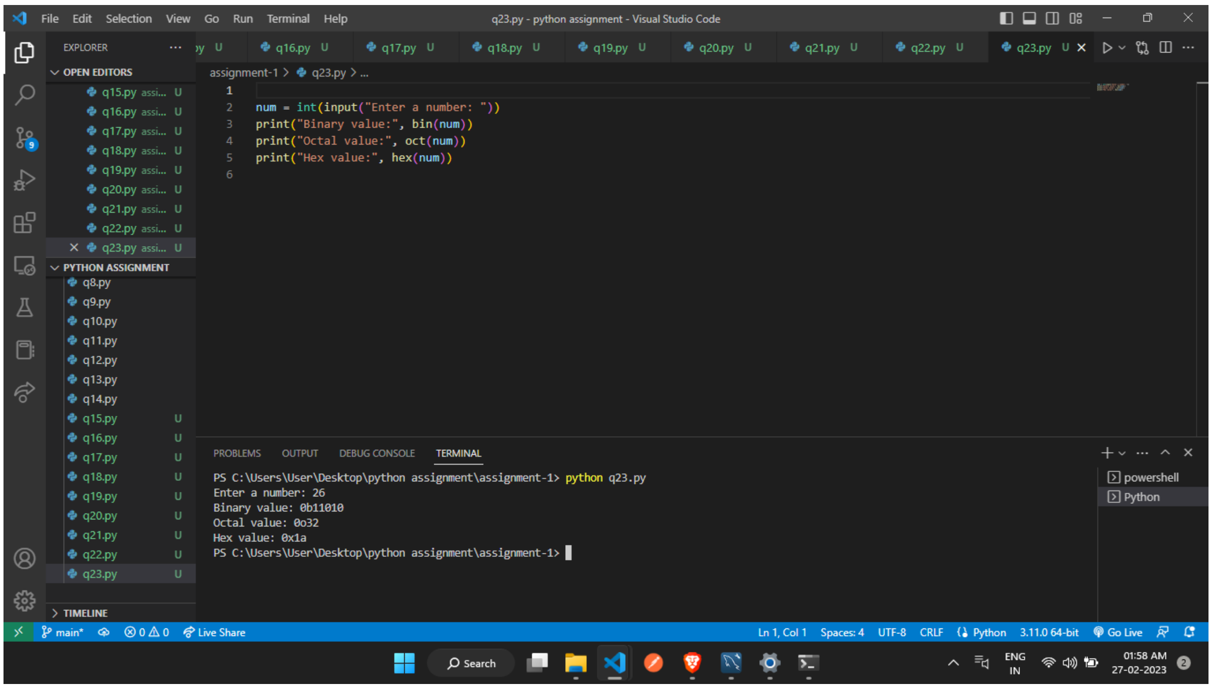


```python

time_in_seconds = int(input("Enter the time in seconds: "))
hours = time_in_seconds // 3600
minutes = (time_in_seconds % 3600) // 60
seconds = time_in_seconds % 60
formatted_time = "{:02d}:{:02d}:{:02d}".format(hours, minutes, seconds)
print("The time in HH:MM:SS format is", formatted_time)
```

```
Enter the x-coordinate of the point on the circumference: 6
Enter the y-coordinate of the point on the circumference: 5
The area of the circle is: 37.69911184307751
PS C:\Users\User\Desktop\python assignment> python q7.py
Enter the time in seconds: 45
The time in HH:MM:SS format is 00:00:45
PS C:\Users\User\Desktop\python assignment> python q7.py
Enter the time in seconds: 12000
The time in HH:MM:SS format is 03:20:00
PS C:\Users\User\Desktop\python assignment>
```

- **Use built in functions**

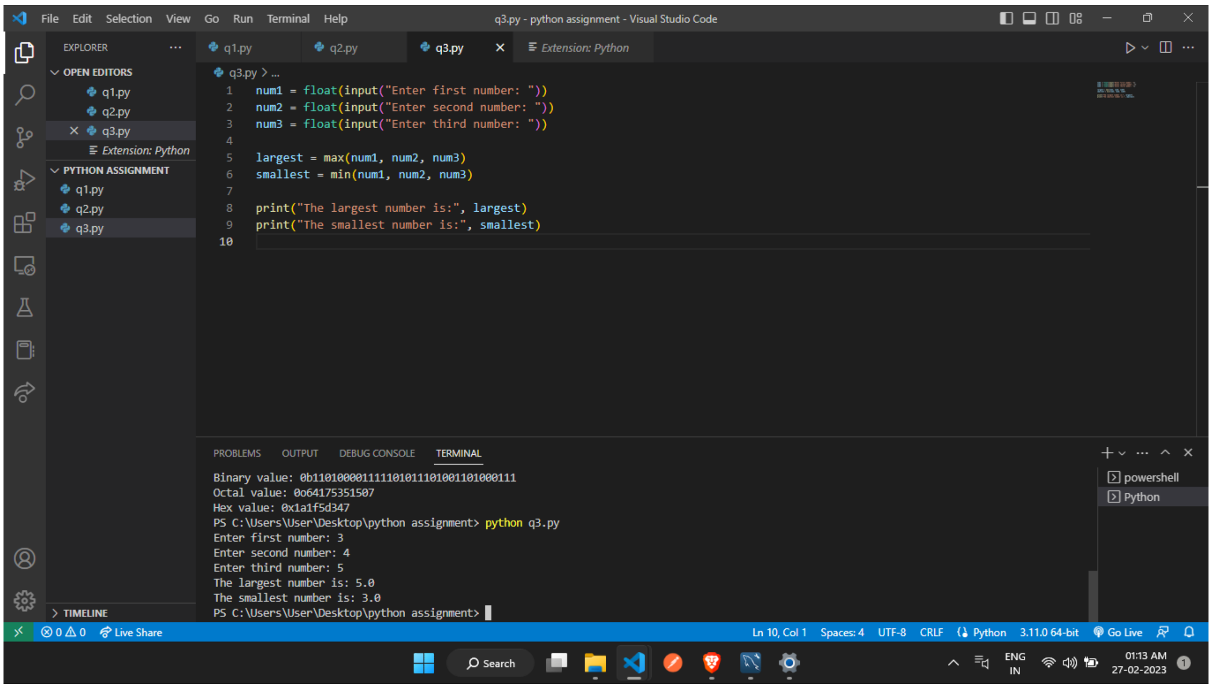
1. **Read a number and Print the corresponding binary, oct, hex**



```python

num = int(input("Enter a number: "))
print("Binary value:", bin(num))
print("Octal value:", oct(num))
print("Hex value:", hex(num))

```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q23.py
Enter a number: 26
Binary value: 0b11010
Octal value: 0o32
Hex value: 0x1a
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

2. **Read 3 numbers and find the largest and smallest**

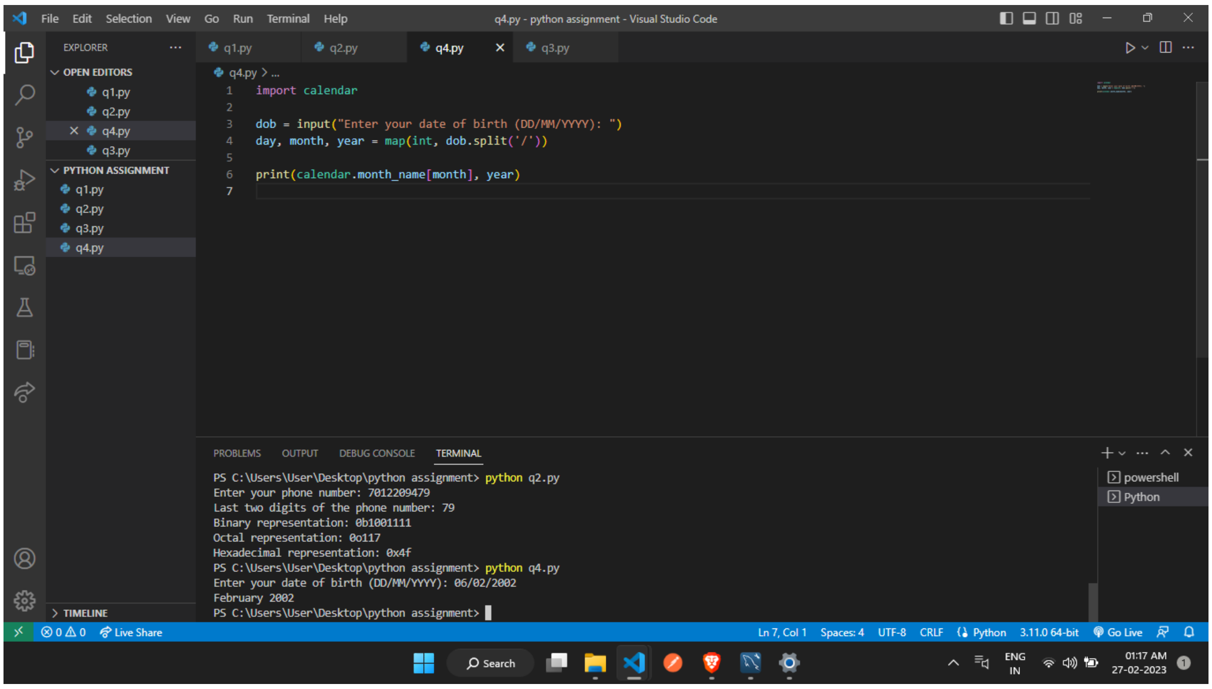


```python
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

largest = max(num1, num2, num3)
smallest = min(num1, num2, num3)

print("The largest number is:", largest)
print("The smallest number is:", smallest)

```

```
Binary value: 0b110100001111101011101001101000111
Octal value: 0o64175351507
Hex value: 0x1a1f5d347
PS C:\Users\User\Desktop\python assignment> python q3.py
Enter first number: 3
Enter second number: 4
Enter third number: 5
The largest number is: 5.0
The smallest number is: 3.0
PS C:\Users\User\Desktop\python assignment>
```

- **Use modules/packages**

1. **Print the month calendar depending on your DOB.**
   Eg: if the dob is 10/10/2004 the program should print 2004 October month calendar

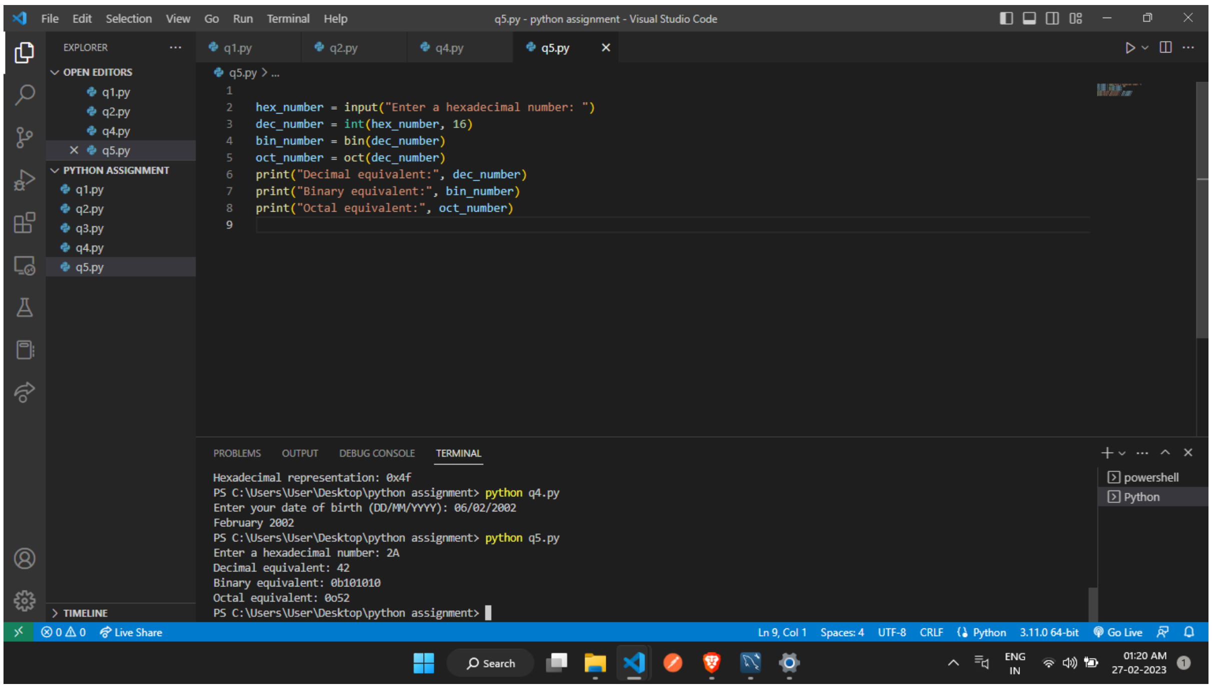


q4.py - python assignment - Visual Studio Code

```python
import calendar

dob = input("Enter your date of birth (DD/MM/YYYY): ")
day, month, year = map(int, dob.split('/'))

print(calendar.month_name[month], year)
```

```
PS C:\Users\User\Desktop\python assignment> python q2.py
Enter your phone number: 7012209479
Last two digits of the phone number: 79
Binary representation: 0b1001111
Octal representation: 0o117
Hexadecimal representation: 0x4f
PS C:\Users\User\Desktop\python assignment> python q4.py
Enter your date of birth (DD/MM/YYYY): 06/02/2002
February 2002
PS C:\Users\User\Desktop\python assignment>
```

2. **Find the number of digits in the factorial of a given number**

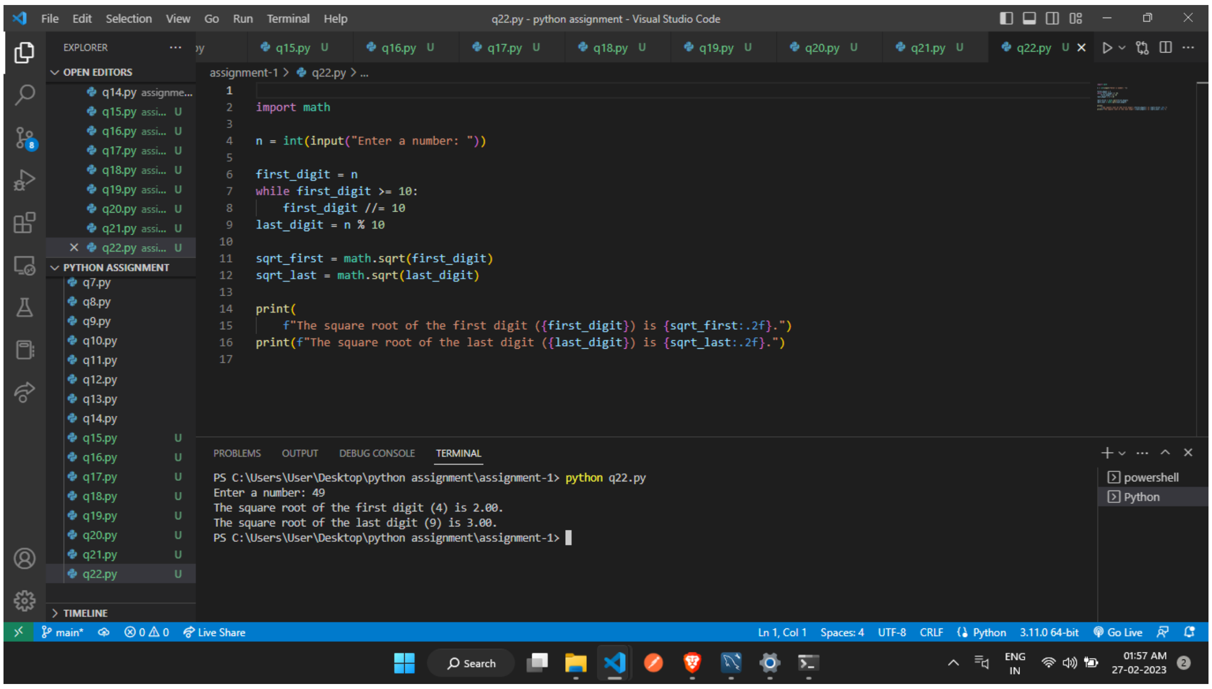


q5.py - python assignment - Visual Studio Code

```python

hex_number = input("Enter a hexadecimal number: ")
dec_number = int(hex_number, 16)
bin_number = bin(dec_number)
oct_number = oct(dec_number)
print("Decimal equivalent:", dec_number)
print("Binary equivalent:", bin_number)
print("Octal equivalent:", oct_number)
```

```
Hexadecimal representation: 0x4f
PS C:\Users\User\Desktop\python assignment> python q4.py
Enter your date of birth (DD/MM/YYYY): 06/02/2002
February 2002
PS C:\Users\User\Desktop\python assignment> python q5.py
Enter a hexadecimal number: 2A
Decimal equivalent: 42
Binary equivalent: 0b101010
Octal equivalent: 0o52
PS C:\Users\User\Desktop\python assignment>
```

3. Find the sqrt of first and last digit of a number

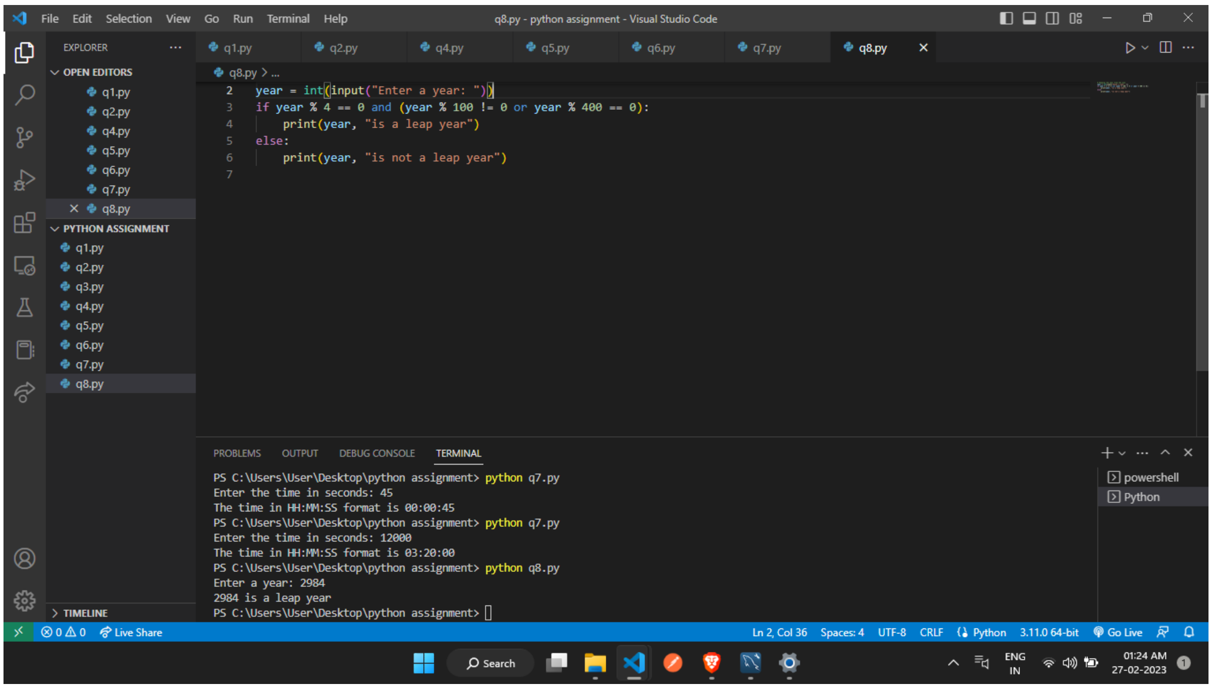


```python
import math

n = int(input("Enter a number: "))

first_digit = n
while first_digit >= 10:
    first_digit //= 10
last_digit = n % 10

sqrt_first = math.sqrt(first_digit)
sqrt_last = math.sqrt(last_digit)

print(
    f"The square root of the first digit ({first_digit}) is {sqrt_first:.2f}.")
print(f"The square root of the last digit ({last_digit}) is {sqrt_last:.2f}.")
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q22.py
Enter a number: 49
The square root of the first digit (4) is 2.00.
The square root of the last digit (9) is 3.00.
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

- **Using if**

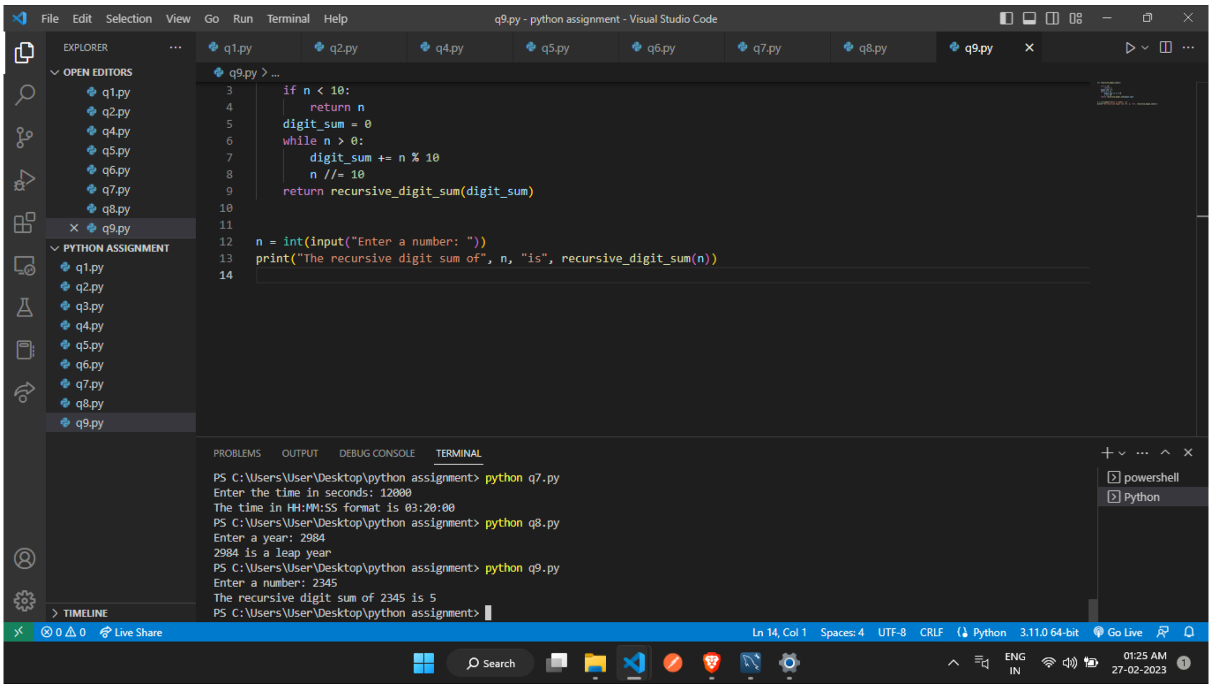
1. Check whether the given year is leap year or not.



```python
year = int(input("Enter a year: "))
if year % 4 == 0 and (year % 100 != 0 or year % 400 == 0):
    print(year, "is a leap year")
else:
    print(year, "is not a leap year")
```

```
PS C:\Users\User\Desktop\python assignment> python q7.py
Enter the time in seconds: 45
The time in HH:MM:SS format is 00:00:45
PS C:\Users\User\Desktop\python assignment> python q7.py
Enter the time in seconds: 12000
The time in HH:MM:SS format is 03:20:00
PS C:\Users\User\Desktop\python assignment> python q8.py
Enter a year: 2984
2984 is a leap year
PS C:\Users\User\Desktop\python assignment>
```

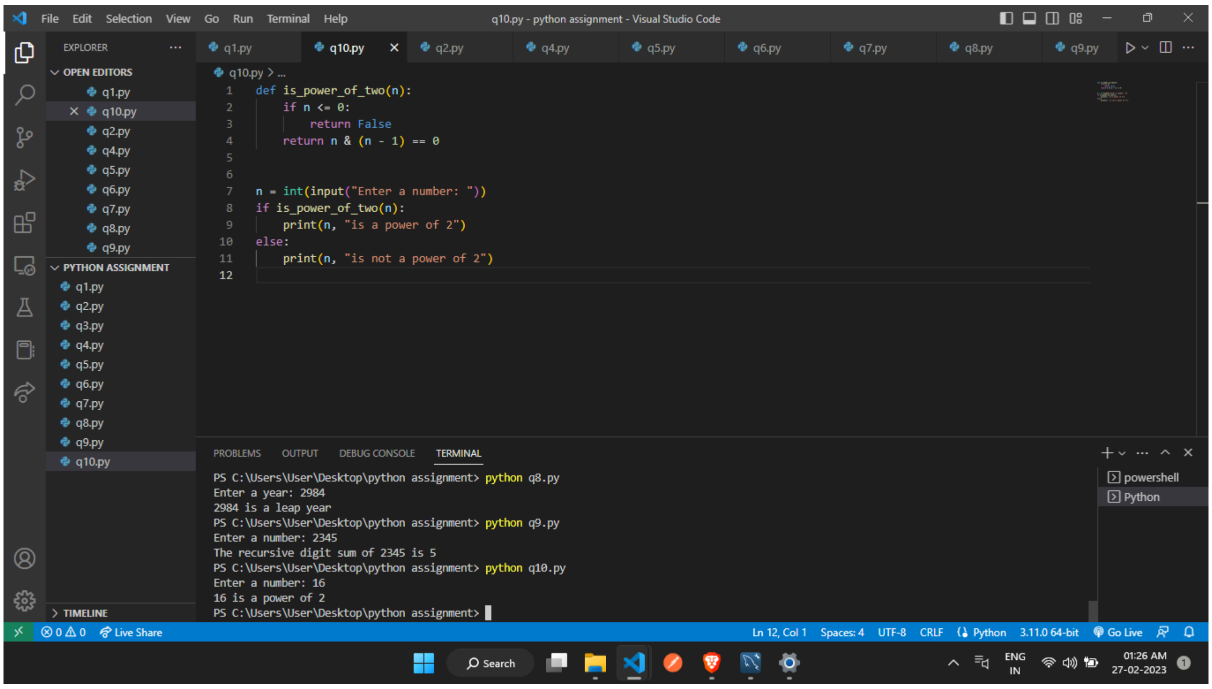
2. **Write a python program to read a number and recursively add the digits in it Eg: i/p:123 o/p:6 i/p:78 o/p:6**



```
    if n < 10:
        return n
    digit_sum = 0
    while n > 0:
        digit_sum += n % 10
        n //= 10
    return recursive_digit_sum(digit_sum)


n = int(input("Enter a number: "))
print("The recursive digit sum of", n, "is", recursive_digit_sum(n))
```

```
PS C:\Users\User\Desktop\python assignment> python q7.py
Enter the time in seconds: 12000
The time in HH:MM:SS format is 03:20:00
PS C:\Users\User\Desktop\python assignment> python q8.py
Enter a year: 2984
2984 is a leap year
PS C:\Users\User\Desktop\python assignment> python q9.py
Enter a number: 2345
The recursive digit sum of 2345 is 5
PS C:\Users\User\Desktop\python assignment>
```

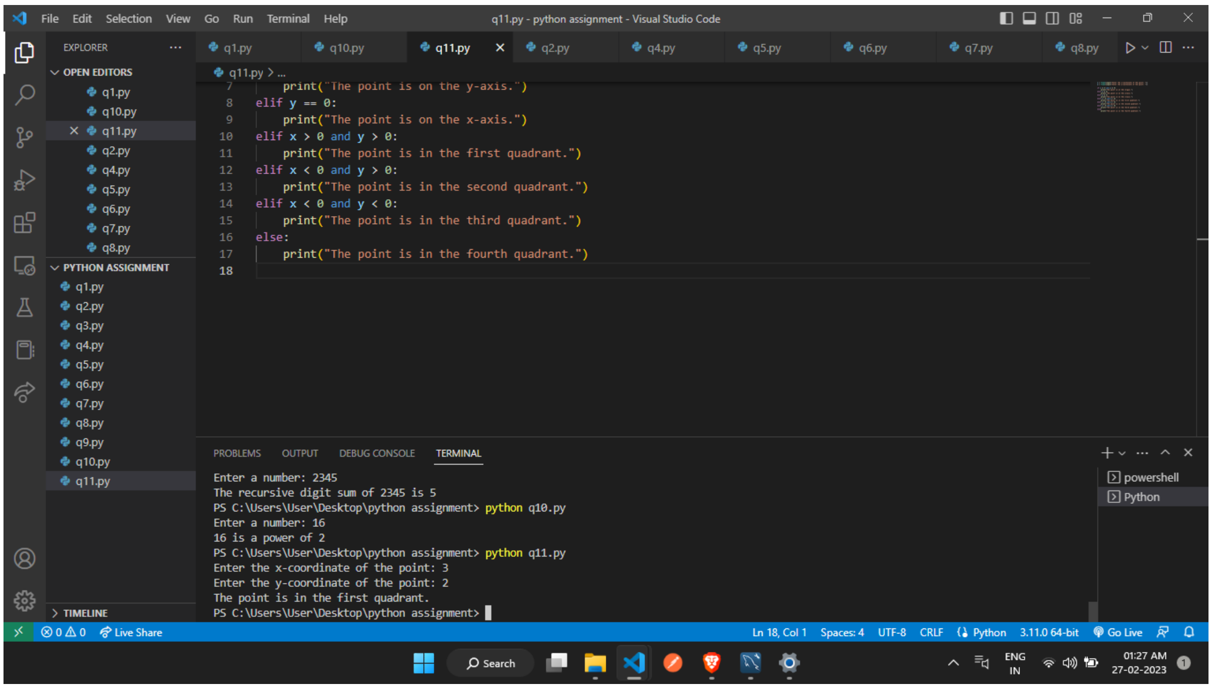
3. **Check whether the given number is poser of 2 (use bitwise operator)**



```
def is_power_of_two(n):
    if n <= 0:
        return False
    return n & (n - 1) == 0


n = int(input("Enter a number: "))
if is_power_of_two(n):
    print(n, "is a power of 2")
else:
    print(n, "is not a power of 2")
```

```
PS C:\Users\User\Desktop\python assignment> python q8.py
Enter a year: 2984
2984 is a leap year
PS C:\Users\User\Desktop\python assignment> python q9.py
Enter a number: 2345
The recursive digit sum of 2345 is 5
PS C:\Users\User\Desktop\python assignment> python q10.py
Enter a number: 16
16 is a power of 2
PS C:\Users\User\Desktop\python assignment>
```

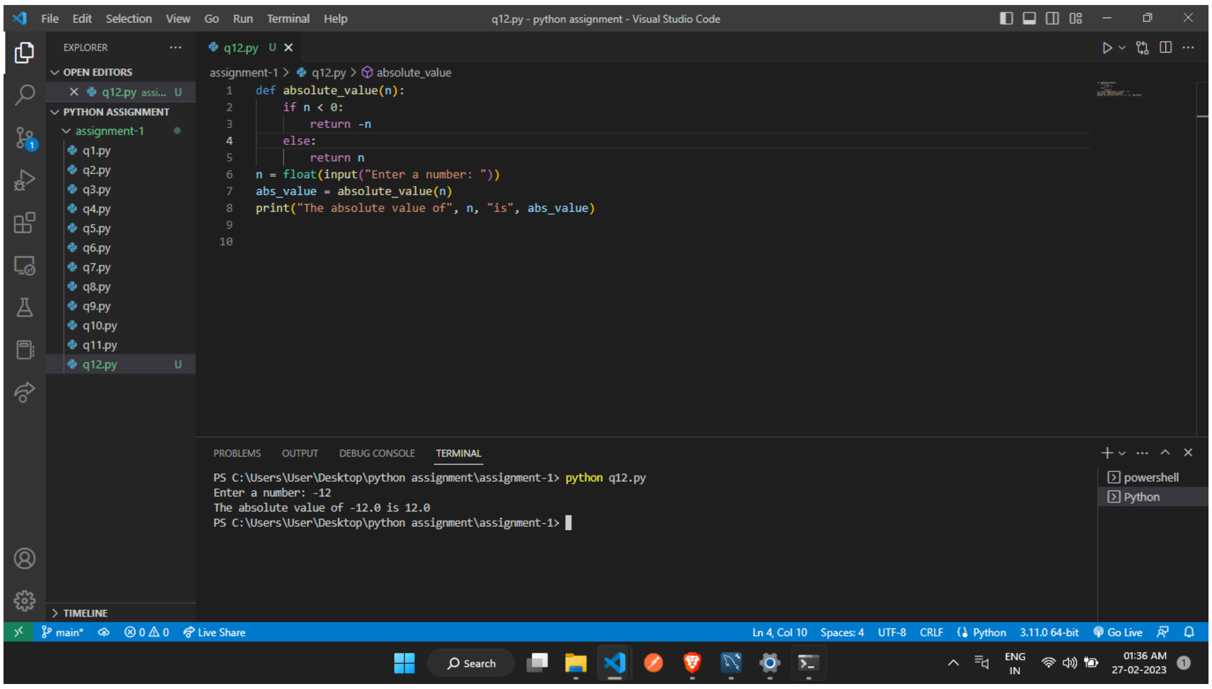
4. **Write a program to check the quadrant of a given point(x,y)(University question)**



```
    print("The point is on the y-axis.")
elif y == 0:
    print("The point is on the x-axis.")
elif x > 0 and y > 0:
    print("The point is in the first quadrant.")
elif x < 0 and y > 0:
    print("The point is in the second quadrant.")
elif x < 0 and y < 0:
    print("The point is in the third quadrant.")
else:
    print("The point is in the fourth quadrant.")
```

```
Enter a number: 2345
The recursive digit sum of 2345 is 5
PS C:\Users\User\Desktop\python assignment> python q10.py
Enter a number: 16
16 is a power of 2
PS C:\Users\User\Desktop\python assignment> python q11.py
Enter the x-coordinate of the point: 3
Enter the y-coordinate of the point: 2
The point is in the first quadrant.
PS C:\Users\User\Desktop\python assignment>
```

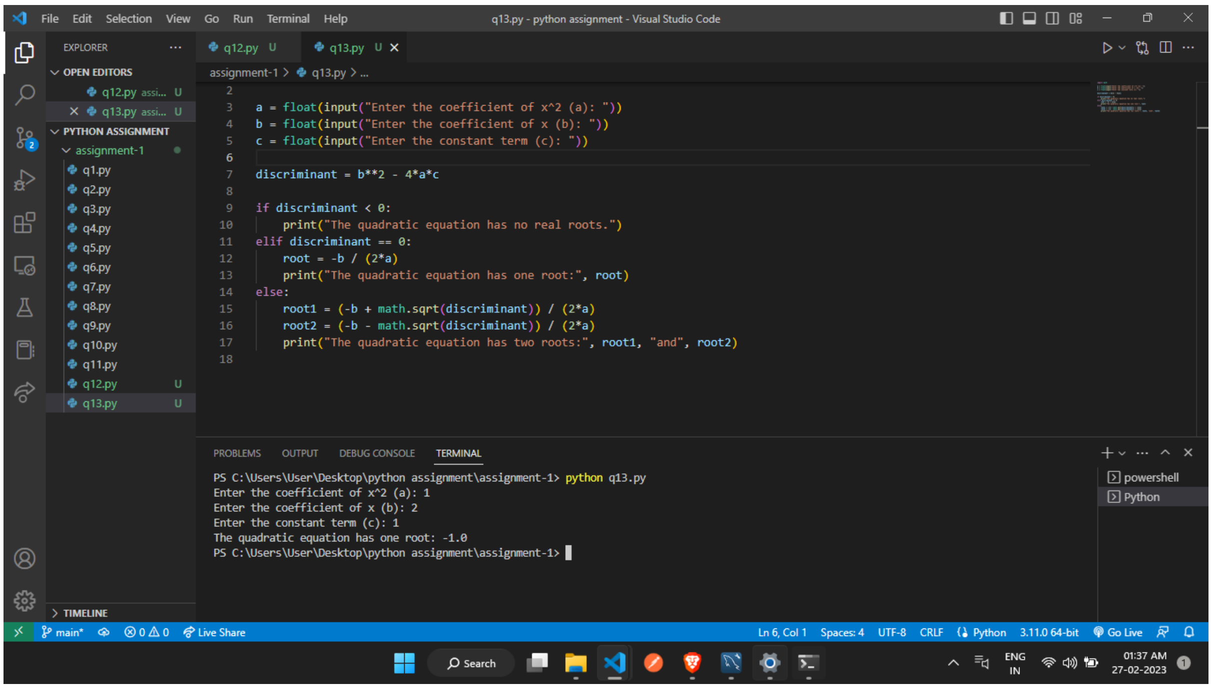
5. **Write a program to get the absolute value of a number without using the abs() function.(university question)**



```
def absolute_value(n):
    if n < 0:
        return -n
    else:
        return n
n = float(input("Enter a number: "))
abs_value = absolute_value(n)
print("The absolute value of", n, "is", abs_value)
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q12.py
Enter a number: -12
The absolute value of -12.0 is 12.0
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

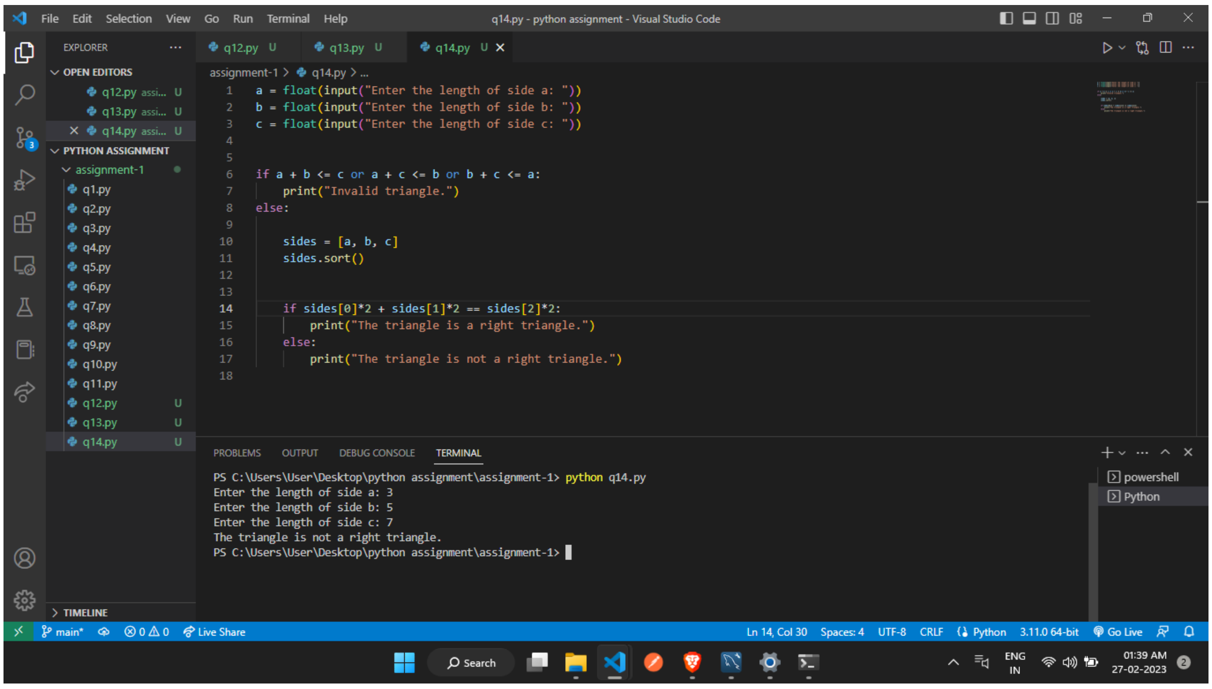
6. **Find the roots of a quadratic equation**



```
a = float(input("Enter the coefficient of x^2 (a): "))
b = float(input("Enter the coefficient of x (b): "))
c = float(input("Enter the constant term (c): "))

discriminant = b**2 - 4*a*c

if discriminant < 0:
    print("The quadratic equation has no real roots.")
elif discriminant == 0:
    root = -b / (2*a)
    print("The quadratic equation has one root:", root)
else:
    root1 = (-b + math.sqrt(discriminant)) / (2*a)
    root2 = (-b - math.sqrt(discriminant)) / (2*a)
    print("The quadratic equation has two roots:", root1, "and", root2)
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q13.py
Enter the coefficient of x^2 (a): 1
Enter the coefficient of x (b): 2
Enter the constant term (c): 1
The quadratic equation has one root: -1.0
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

7. **Write a program that accepts the length of three sides of a triangle as input and determine whether ornot the triangle is a right triangle.(university question)**

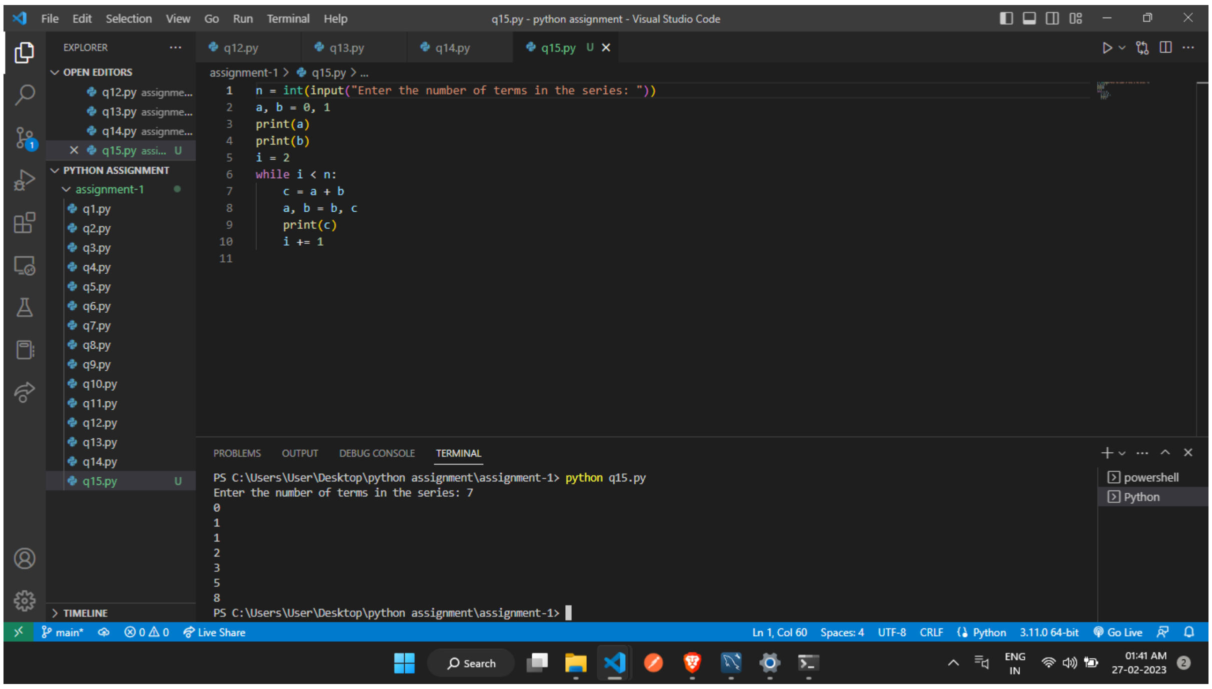


```
a = float(input("Enter the length of side a: "))
b = float(input("Enter the length of side b: "))
c = float(input("Enter the length of side c: "))


if a + b <= c or a + c <= b or b + c <= a:
    print("Invalid triangle.")
else:

    sides = [a, b, c]
    sides.sort()


    if sides[0]*2 + sides[1]*2 == sides[2]*2:
        print("The triangle is a right triangle.")
    else:
        print("The triangle is not a right triangle.")
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q14.py
Enter the length of side a: 3
Enter the length of side b: 5
Enter the length of side c: 7
The triangle is not a right triangle.
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

- **Using while/for**

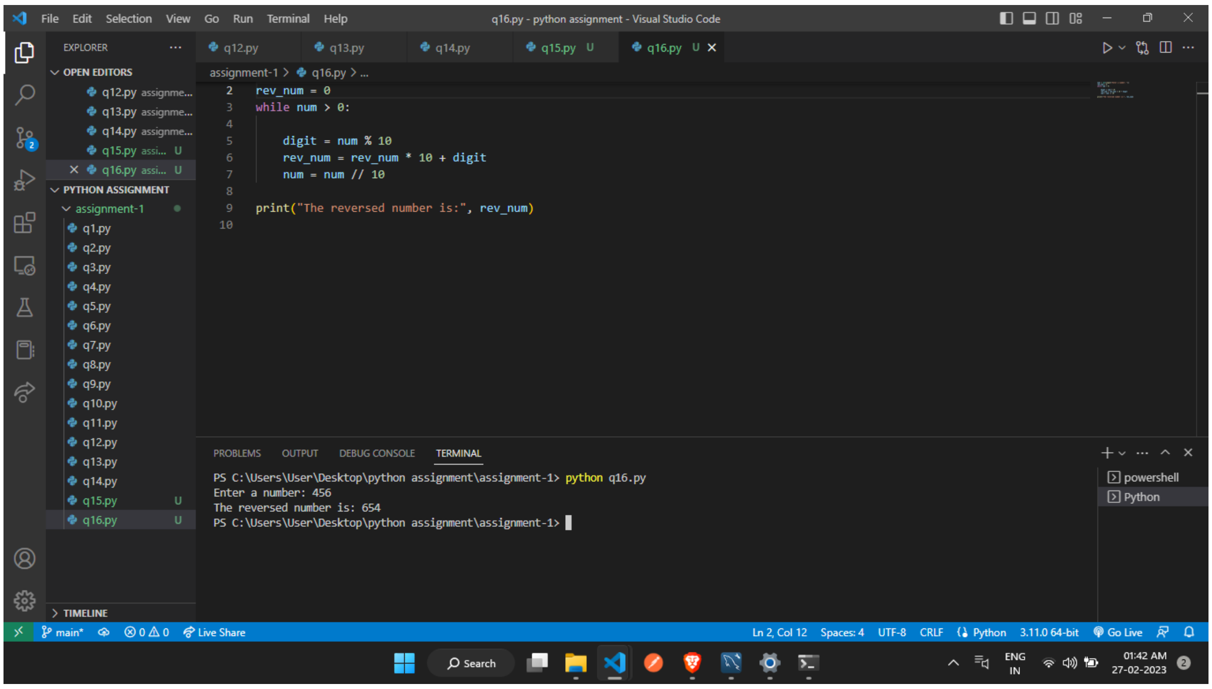
1. **Generate the Fibonacci series 0 1 1 2 3 5 8…..n (use while..read n)**



```python
n = int(input("Enter the number of terms in the series: "))
a, b = 0, 1
print(a)
print(b)
i = 2
while i < n:
    c = a + b
    a, b = b, c
    print(c)
    i += 1
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q15.py
Enter the number of terms in the series: 7
0
1
1
2
3
5
8
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

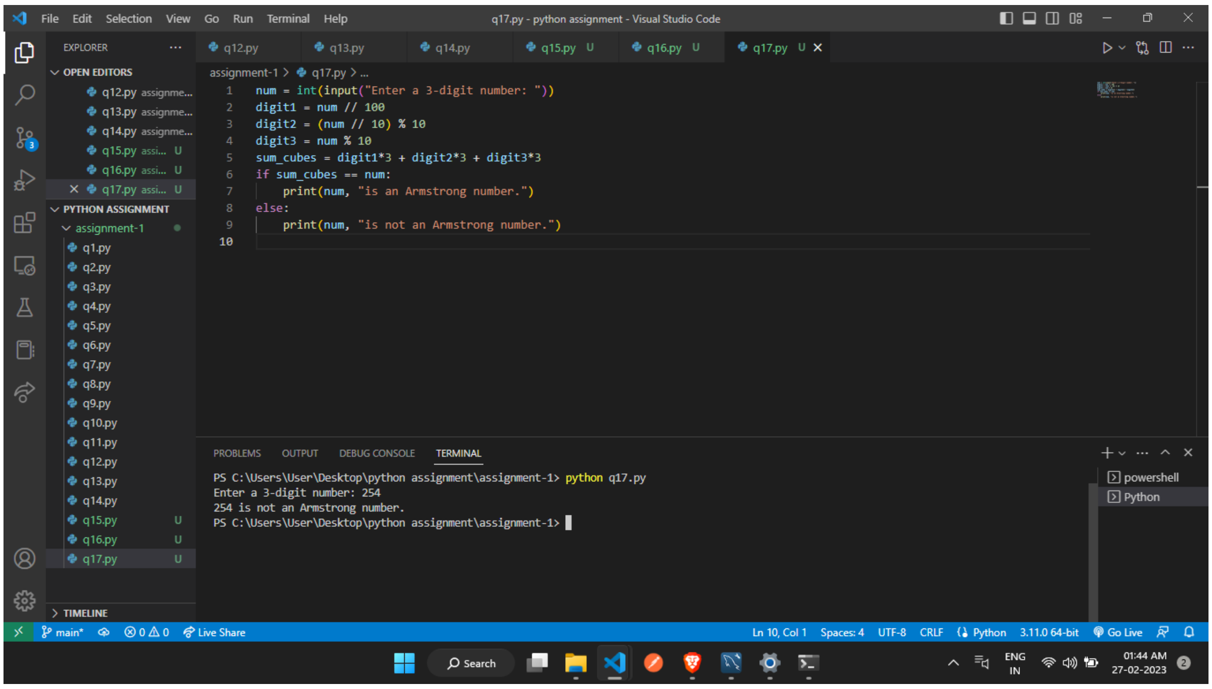
2. **Reverse a Number (i/p: 123 o/P 321)**



```python
rev_num = 0
while num > 0:

    digit = num % 10
    rev_num = rev_num * 10 + digit
    num = num // 10

print("The reversed number is:", rev_num)
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q16.py
Enter a number: 456
The reversed number is: 654
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

3. Check whether the given 3 digit number is Armstrong Number.

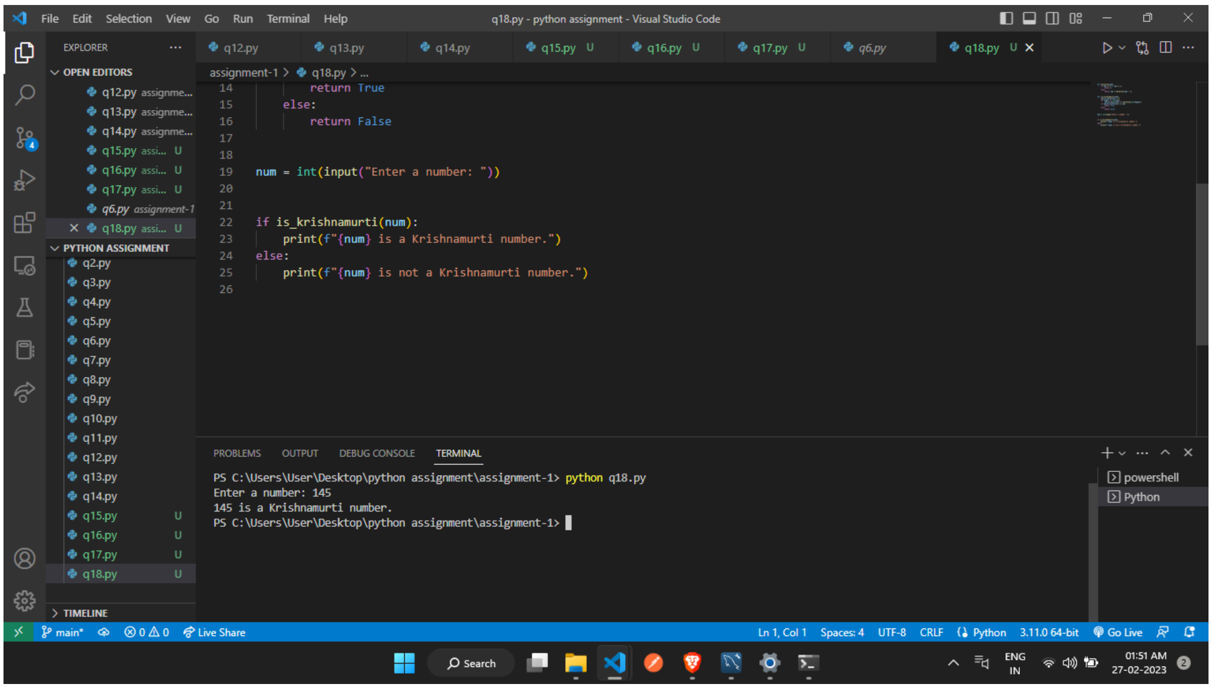


```python
num = int(input("Enter a 3-digit number: "))
digit1 = num // 100
digit2 = (num // 10) % 10
digit3 = num % 10
sum_cubes = digit1*3 + digit2*3 + digit3*3
if sum_cubes == num:
    print(num, "is an Armstrong number.")
else:
    print(num, "is not an Armstrong number.")

```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q17.py
Enter a 3-digit number: 254
254 is not an Armstrong number.
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

4. Check whether the given number is a Krishnamurti number(Krishnamurthy Number: It is a number which is equal to the sum of the factorials of all its digits. Usefactorial() function from math

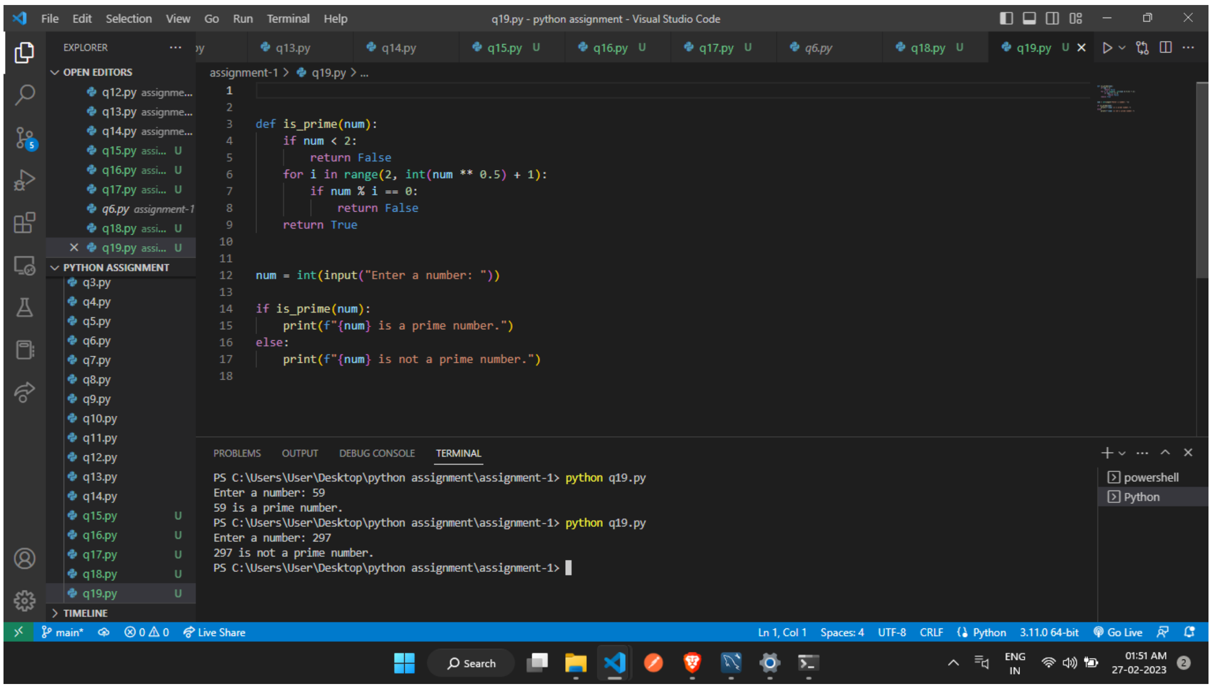
For example : 145 = 1! + 4! + 5! = 1 + 24 + 120 = 145 which is a Krishnamurti Number.



```python
            return True
        else:
            return False


num = int(input("Enter a number: "))


if is_krishnamurti(num):
    print(f"{num} is a Krishnamurti number.")
else:
    print(f"{num} is not a Krishnamurti number.")

```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q18.py
Enter a number: 145
145 is a Krishnamurti number.
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

5. **Check whether the given number is Prime or not**

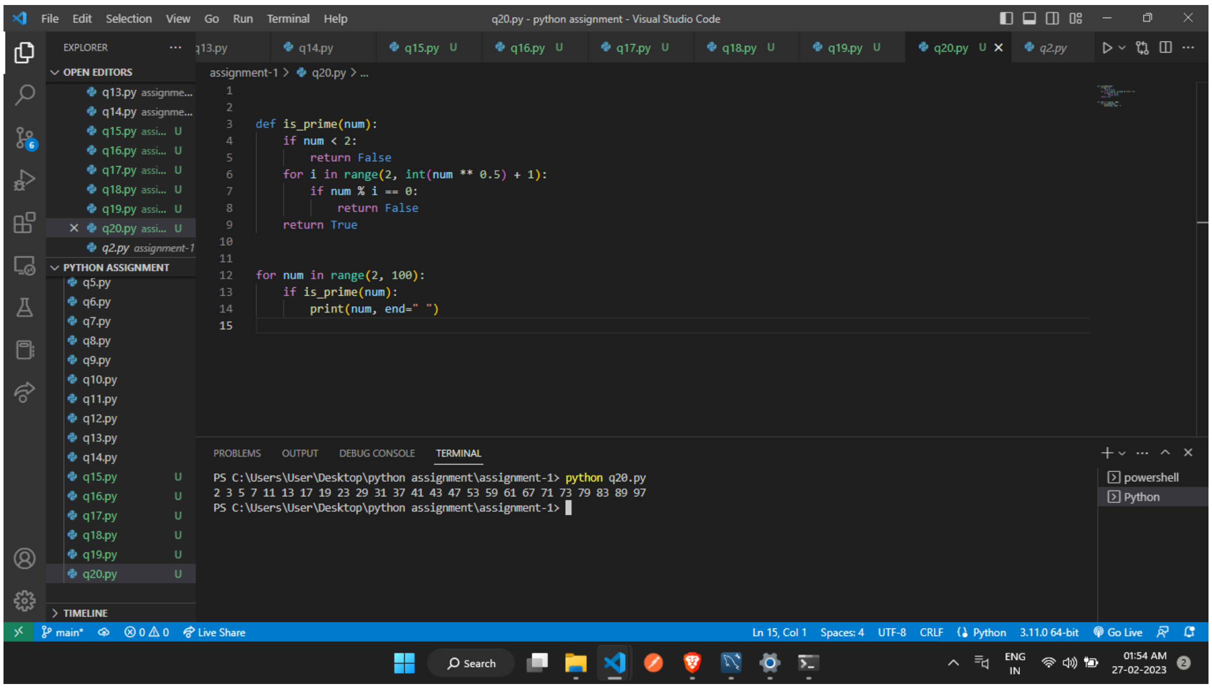

q19.py - python assignment - Visual Studio Code
assignment-1 > q19.py > ...

```python
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True


num = int(input("Enter a number: "))

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

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q19.py
Enter a number: 59
59 is a prime number.
PS C:\Users\User\Desktop\python assignment\assignment-1> python q19.py
Enter a number: 297
297 is not a prime number.
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

- **Nested Loops(for/while)**

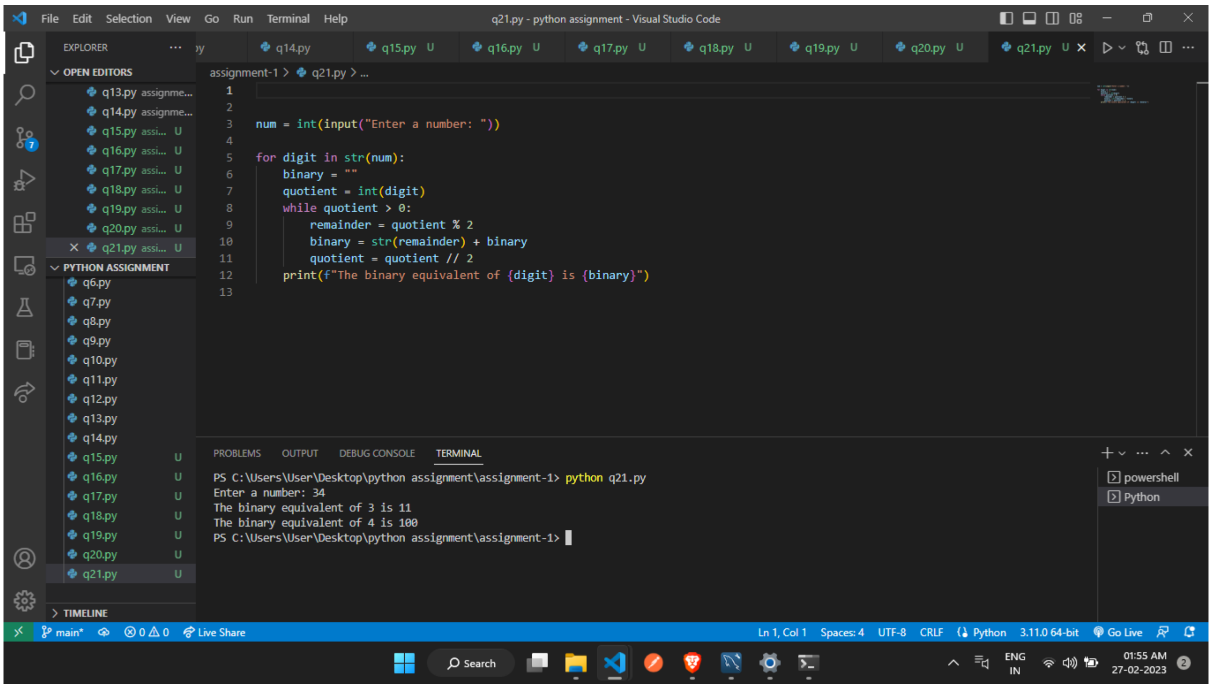
1. **Print all prime numbers less than 100.**


q20.py - python assignment - Visual Studio Code
assignment-1 > q20.py > ...

```python
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True


for num in range(2, 100):
    if is_prime(num):
        print(num, end=" ")
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q20.py
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
PS C:\Users\User\Desktop\python assignment\assignment-1>
```

2. Print the binary equivalent of each digit of the given number



```python

num = int(input("Enter a number: "))

for digit in str(num):
    binary = ""
    quotient = int(digit)
    while quotient > 0:
        remainder = quotient % 2
        binary = str(remainder) + binary
        quotient = quotient // 2
    print(f"The binary equivalent of {digit} is {binary}")
```

```
PS C:\Users\User\Desktop\python assignment\assignment-1> python q21.py
Enter a number: 34
The binary equivalent of 3 is 11
The binary equivalent of 4 is 100
PS C:\Users\User\Desktop\python assignment\assignment-1>
```