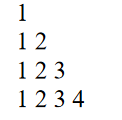
ASSIGNMENT I

1. List identity operators in Python

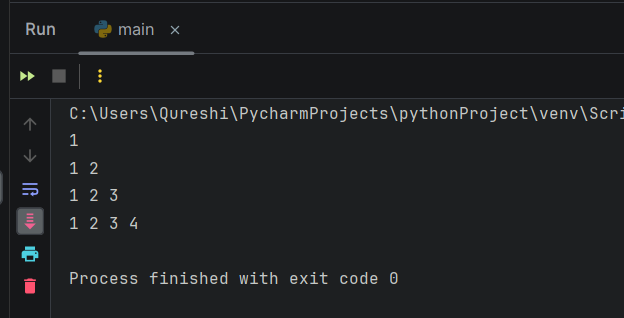
* is: Returns True if both variables point to the same object.
* is not: Returns True if both variables do not point to the same object.

1.  Write a program to print following:

Code:

for i in range(1, 5):  
 for j in range(1, i + 1):  
 print(j, end=" ")  
 print()

Output:



1. Explain membership and assignment operators with example

Membership operators in Python:

* in: Returns True if a value exists in a sequence, else False.
* not in: Returns True if a value does not exist in a sequence, else False.

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| in | True if value is found in list or in sequence, and false it item is not in list or in sequence | >>> x="Hello World"  >>> print('H' in x)  True |
| Not in | True if value is not found in list or in sequence, and false id item is in list or in sequence | >>> x="Hello World" >>> print("Hello” not in x) false |

ASSIGNMENT OPERATORS

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| = | Assigns values from right side operands to left side operand. | c = a+b assigns value of a + b into c |
| += | It adds right operand to the left operand and assign the result to left operand. | c +=a is equivalent to c = c + a |
| -= | It subtracts right operand from the left operand and assign the result to left operand. | c -=a is equivalent to c = c - a |
| \*= | It multiplies right operand with the left operand and assign the result to left operand. | c \* = a is equivalent to c =c \* a |
| /= | It divides left operand with the right operand and assign the result to left operand. | c/= a is equivalent to c = c / a |
| %= | It takes modulus using two operands and assign the result to left operand. | c% = a is equivalent to c % a = a |
| \*\*= | Performs exponential (power) calculation on operators and assign value to the left operand. | c \*\* =a is equivalent to c = c\*\*a |
| //= | Performs exponential (power) calculation on operators and assign value to the left operand. | c//= a is equivalent to c =c//a |

1. Explain decision making statements If- else, if- elif- else with example.

* if-else statement: Executes a block of code if the condition is True, otherwise executes another block of code.
* if-elif-else statement: Executes different blocks of code based on different conditions.

Code:

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

if num > 0:

print("Positive number")

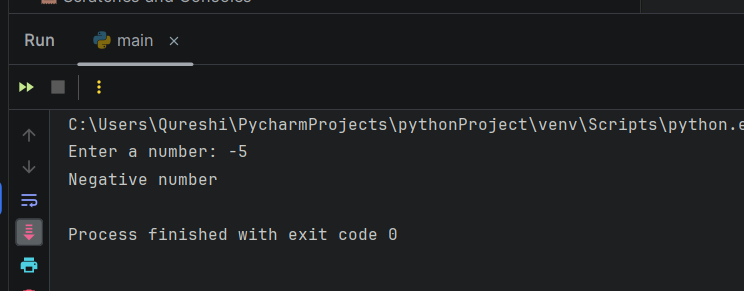
elif num == 0:

print("Zero")

else:

print("Negative number")

Output:



if-elif-else (ladder) statements: Here, a user can decide among multiple options. The if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.

Code:

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

if num > 0:

print("Positive number")

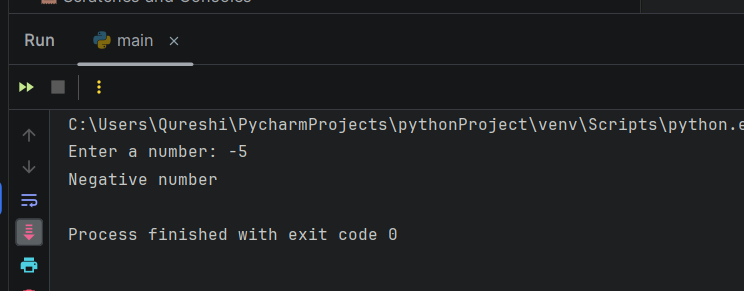
elif num == 0:

print("Zero")

else:

print("Negative number")

Output:



1. List comparison operators in Python.

|  |  |
| --- | --- |
| Operators | Meaning |
| == | Equal to |
| != | Not Equal to |
| < | Less than |
| > | Greater than |
| <= | Less than equal to |
| >= | Greater than equal to |

1. Describe bitwise operators in Python with example.

Bitwise operators acts on bits and performs bit by bit operation.

Assume a=10 (1010) and b=4 (0100)

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Meaning | Description | Example |
| & | Bitwise AND | This operation performs AND Operation between operands.  Operator copies a bit, to the result, if it exists in both operands | a &b = 1010 & 0100 = 0000 =0 |
| | | Bitwise OR | This operation performs OR operation between operands. It copies a bit, if it exists in either operand. | a|b = 1010 | 0100 = 1110 = 14 |
| ^ | Bitwise XOR | This operation performs XOR operations between operands. It copies the bit, if it is set in one operand but not both. | a^b=1010 ^ 0100 = 1110 =14 |
| ~ | Bitwise One’s Complement | It is unary operator and has the effect of  'flipping' bits i.e. opposite the bits of operand. | ~a= ~ 1010 = 0101 |
| >> | Binary left shift | The left operand's value is moved right by the number of bits specified by the right operand. | a>>2 = 1010 >> 2 =0010 = 2 |
| << | Binary right shift | The left operand'svalue is moved left by the number of bits specified by the right operand. | a<<2 =1010 << 2 =101000 = 40 |

7) Write python program to illustrate if else ladder

Code:

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

if num > 0:

print("Positive number")

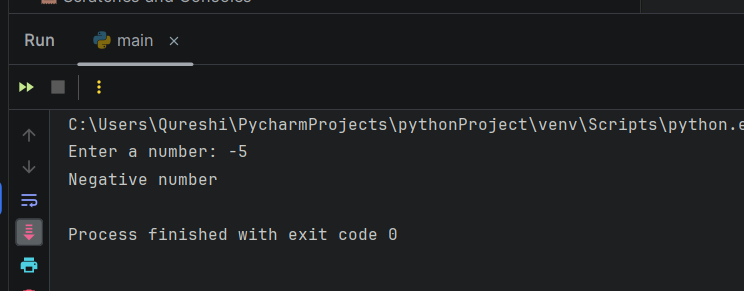
elif num == 0:

print("Zero")

else:

print("Negative number")

Output:



1. Describe membership operators in python

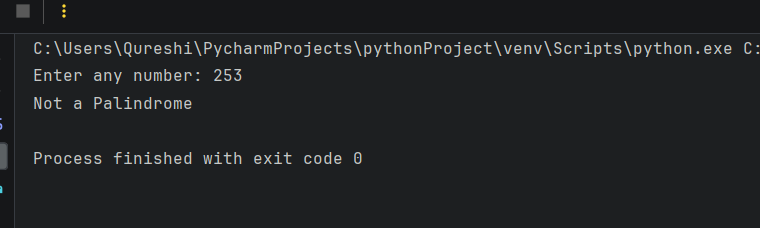
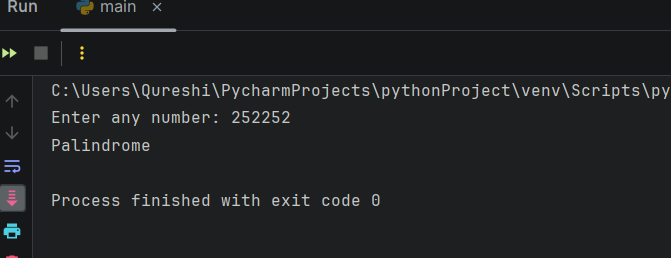
Membership operators (in and not in) are used to test if a value or variable is found in a sequence (string, list, tuple, set, dictionary).

1. Check whether a string is palindrome or not.

Code:

rev = 0  
n = int(input("Enter any number: "))  
i = n  
while n != 0:  
 d = n % 10  
 rev = rev \* 10 + d  
 n = int(n / 10)  
  
  
if i == rev:  
 print("Palindrome")  
else:  
 print("Not a Palindrome")

Output:

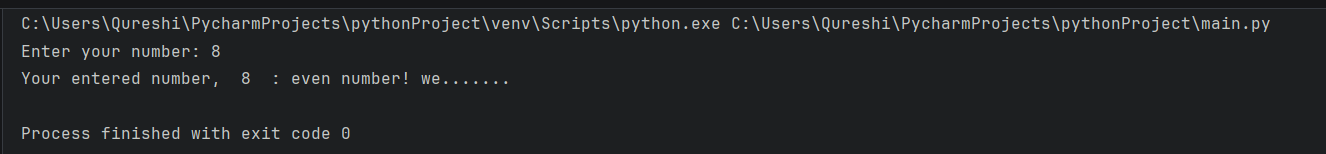
**** ****

11) Check whether a number is even or odd.

Code:

inputNum = int(input("Enter your number: "))  
if inputNum % 2 == 0:  
 print("Your entered number, ", inputNum, " : even number! we.......")  
else:  
 print("Your entered number, ", inputNum, " : odd number! bruh")

Output:

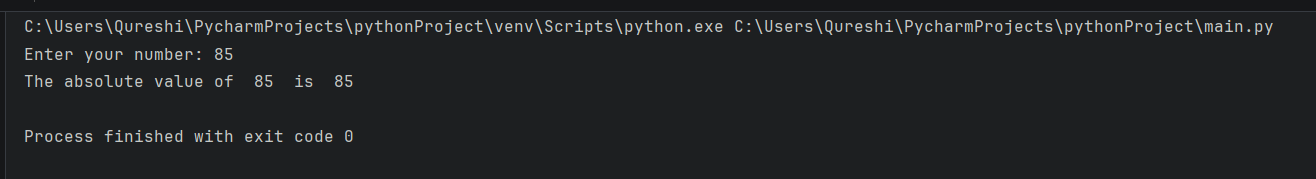
****

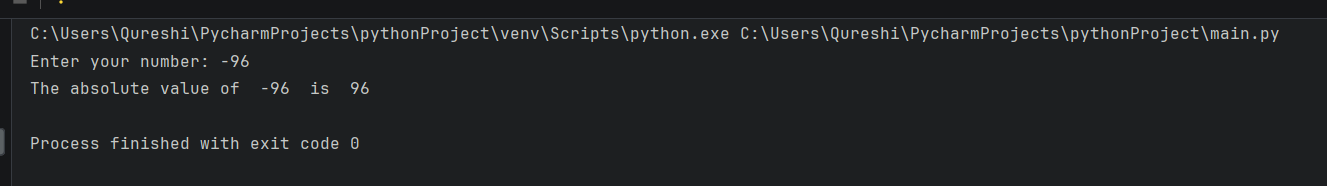
12) Find out absolute value of an input number.

Code:

inputNum = int(input("Enter your number: "))  
absVal = inputNum  
if inputNum < 0:  
 absVal = inputNum \* (-1)  
print("The absolute value of ", inputNum, " is ", absVal)

Output:

****

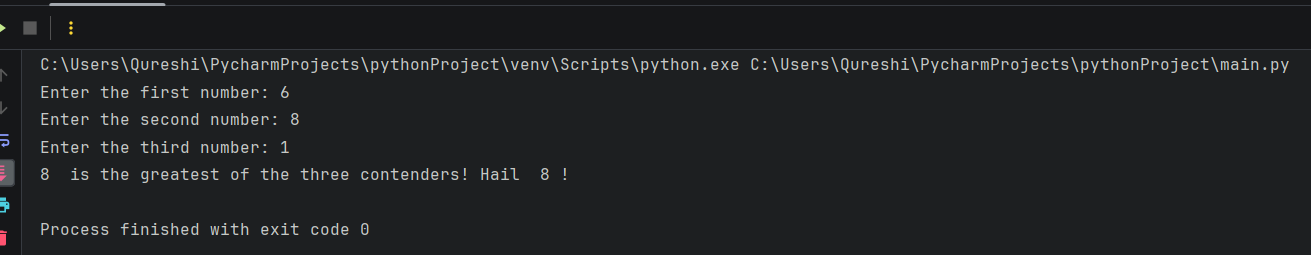
****

12) Check the largest number among three numbers.

Code:

a = int(input("Enter the first number: "))  
b = int(input("Enter the second number: "))  
c = int(input("Enter the third number: "))  
if a > b and a > c:  
 print(a, " is the greatest of the three contenders! Hail ", a, "!")  
elif b > c:  
 print(b, " is the greatest of the three contenders! Hail ", b, "!")  
else:  
 print(c, " is the greatest of the three contenders! Hail ", c, "!")

Output:

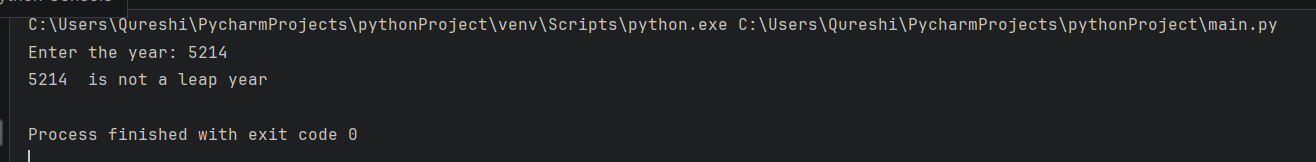
****

13) Check if input year is a leap year or not.

Code:

n = int(input("Enter the year: "))  
if n % 4 == 0:  
 print("Its: ", n, " is a leap year!")  
else:  
 print(n, " is not a leap year")

Output:

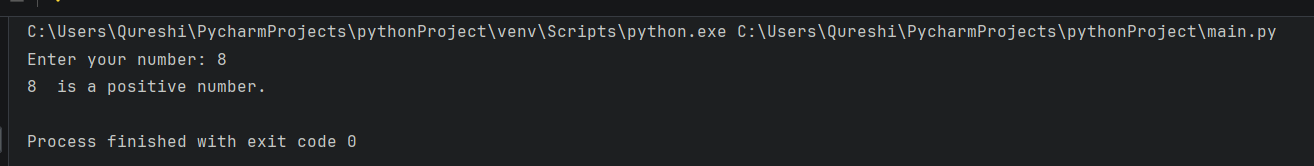
****

14) Check if a number is positive, negative or zero.

Code:

n = int(input("Enter your number: "))  
if n > 0:  
 print(n, " is a positive number.")  
elif n < 0:  
 print(n, " is a negative number.")  
else:  
 print(n, " is zero.")

Output:

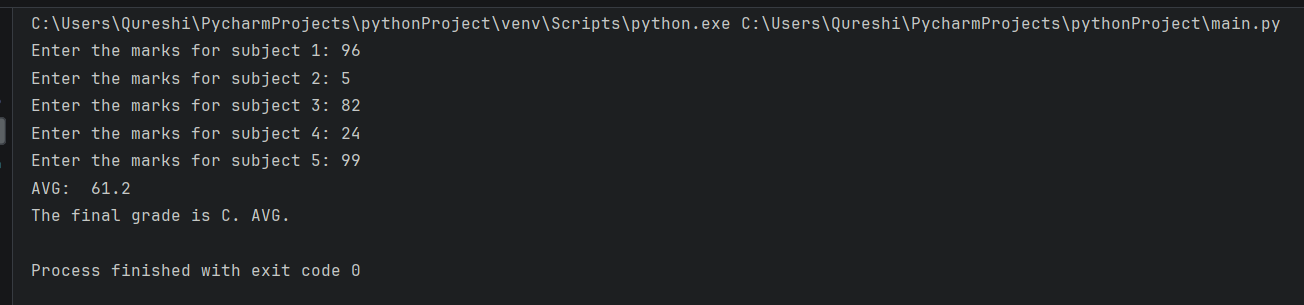
****

15) Program that takes the marks of 5 subjects and displays the grade.

Code:

s1 = float(input("Enter the marks for subject 1: "))  
s2 = float(input("Enter the marks for subject 2: "))  
s3 = float(input("Enter the marks for subject 3: "))  
s4 = float(input("Enter the marks for subject 4: "))  
s5 = float(input("Enter the marks for subject 5: "))  
AVG = (s1 + s2 + s3 + s4 + s5) / 5  
print("AVG: ", AVG)  
if AVG >= 75:  
 print("The final grade is A. Congratulations!")  
elif AVG >= 65:  
 print("The final grade is B. Impressive!")  
elif AVG >= 45:  
 print("The final grade is C. AVG.")  
elif AVG >= 35:  
 print("The final grade is D. Rather poor.")  
else:  
 print("The final grade is F. You have failed.")

Output:

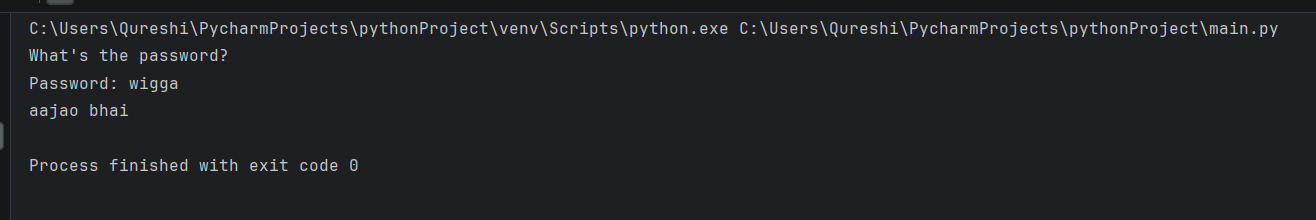
****

16) Check if entered password is correct or not.

Code:

n = input("What's the password?\nPassword: ")  
if n == "wigga":  
 print("aajao bhai")  
else:  
 print("get the hell out")

Output:

****

17) Print the following patterns using loop:

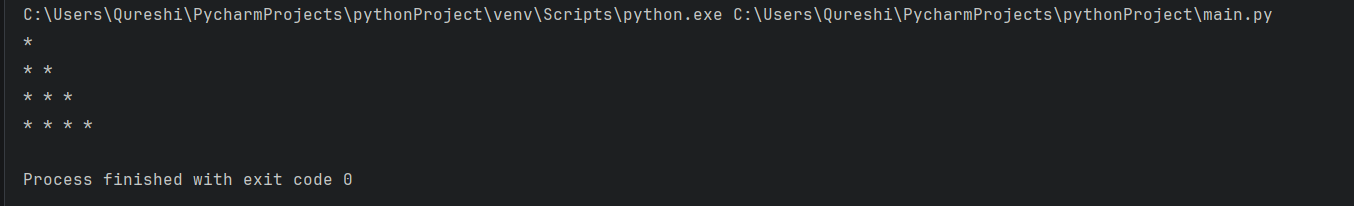
1. **\*  
   \*\*  
   \*\*\*  
   \*\*\*\***

CODE:

for i in range(1, 5):

print('\* ' \* i)

OUTPUT:

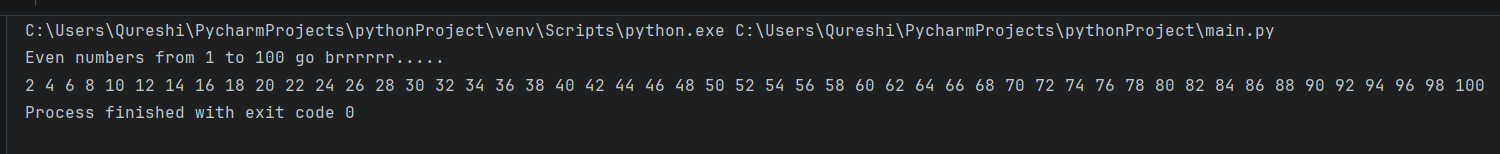
****

18) Print all even number between 1 to 100 using while loop.

**CODE:**

print("Even numbers from 1 to 100 go brrrrrr.....")  
myNum = 1  
while myNum <= 100:  
 if myNum % 2 == 0:  
 print(myNum, end=" ")  
 myNum += 1

**OUTPUT:**

****

19) Find the sum of first 10 natural numbers using for loop.

**CODE:**

print("The sum of the first 10 natural numbers:", end=" ")

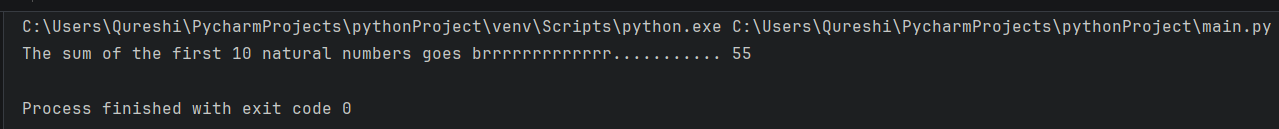
Sum = 0

for i in range(1, 11):

Sum += i

print(Sum)

**OUTPUT:**

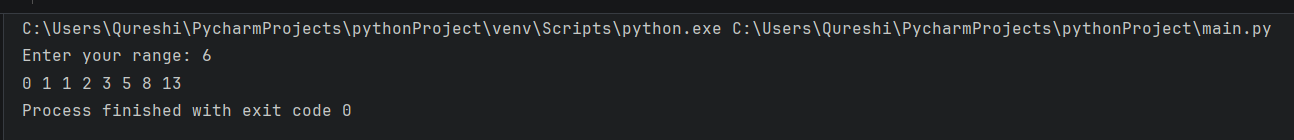
****

20) Print Fibonacci series

**CODE:**

limiter = int(input("Enter your range: "))  
t1 = 0  
t2 = 1  
t3 = 0  
print(t1, t2, end=" ")  
for i in range(limiter):  
 t3 = t1 + t2  
 print(t3, end=" ")  
 t1 = t2  
 t2 = t3

**OUTPUT:**

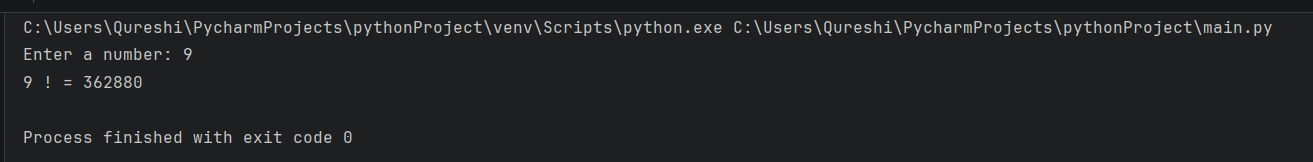
****

21) Calculate factorial of a number.

**CODE:**

n = int(input("Enter a number: "))  
fact = 1  
print(n, "! =", end=" ")  
while n >= 1:  
 fact \*= n  
 n -= 1  
  
print(fact)

**OUTPUT:**

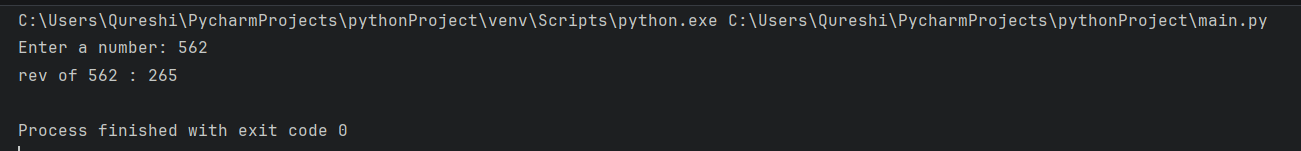
****

22) Reverse a given number.

**CODE:**

n = int(input("Enter a number: "))  
rev = 0  
mod = 0  
temp = n  
while temp > 0:  
 mod = temp % 10  
 rev = (rev \* 10) + mod  
 temp = int(temp / 10)  
print("rev of", n, ":", rev)

**OUTPUT:**

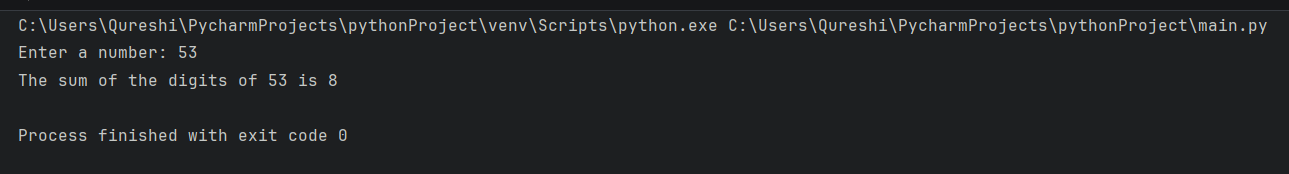
****

23) Program that takes in a number and finds the sum of the digits in the number.

**CODE:**

n = int(input("Enter a number: "))  
Sum = 0  
mod = 0  
temp = n  
while temp > 0:  
 mod = temp % 10  
 Sum += mod  
 temp = int(temp / 10)  
print("The sum of the digits of", n, "is", Sum)

**OUTPUT:**

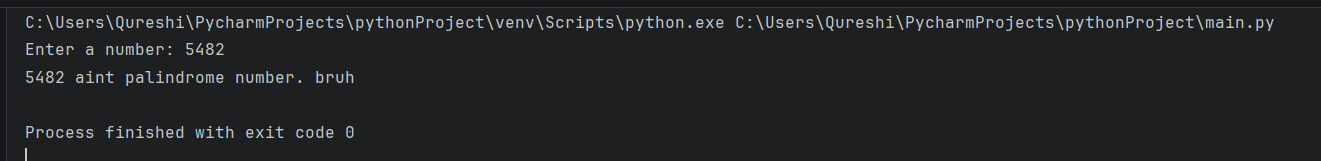
****

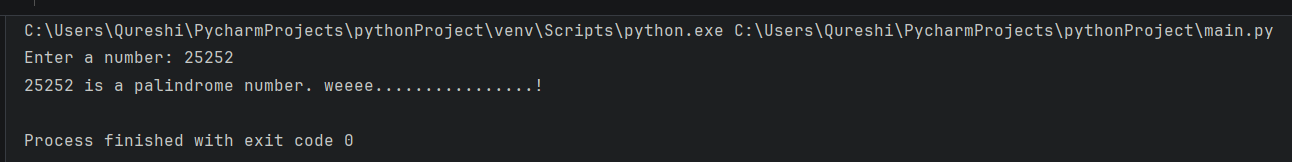
24) Program that takes a number and check whether it is a palindrome.

**CODE:**

n = int(input("Enter a number: "))  
Sum = 0  
mod = 0  
temp = n  
while temp > 0:  
 mod = temp % 10  
 Sum = (Sum \* 10) + mod  
 temp = int(temp / 10)  
if Sum == n:  
 print(n, "is a palindrome number. weeee................!")  
else:  
 print(n, "aint palindrome number. bruh")

**OUTPUT:**

****

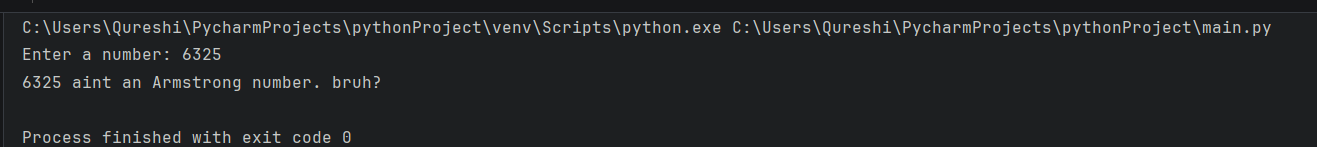
****

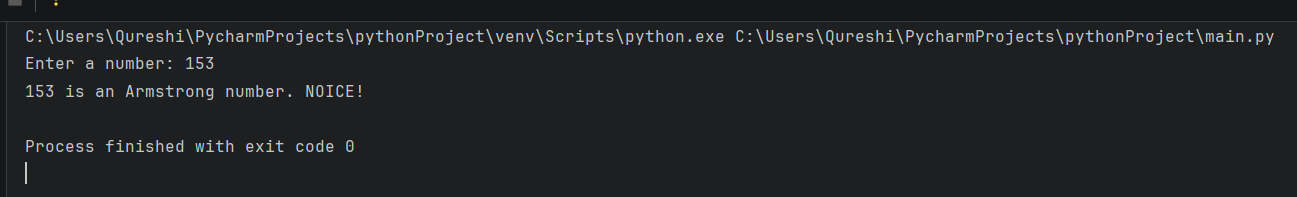
25) Check whether entered number is an Armstrong number.

**CODE:**

n = int(input("Enter a number: "))  
Sum = 0  
temp = n  
mod = 0  
while n > 0:  
 mod = n % 10  
 Sum = Sum + mod \*\* 3  
 n = int(n / 10)  
if Sum == temp:  
 print(temp, "is an Armstrong number. NOICE!")  
else:  
 print(temp, "aint an Armstrong number. bruh?")

**OUTPUT:**

****

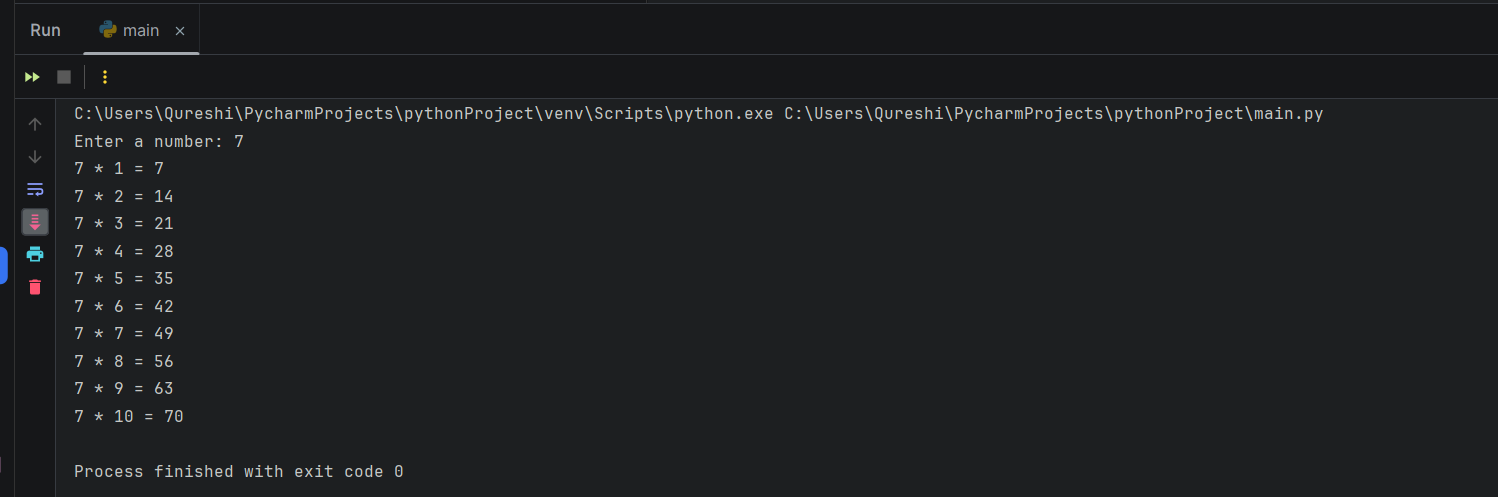
****

26) Print the table of entered number.

**CODE:**

n = int(input("Enter a number: "))  
i = 1  
while i <= 10:  
 print(n, "\*", i, "=", n \* i)  
 i += 1

**OUTPUT:**

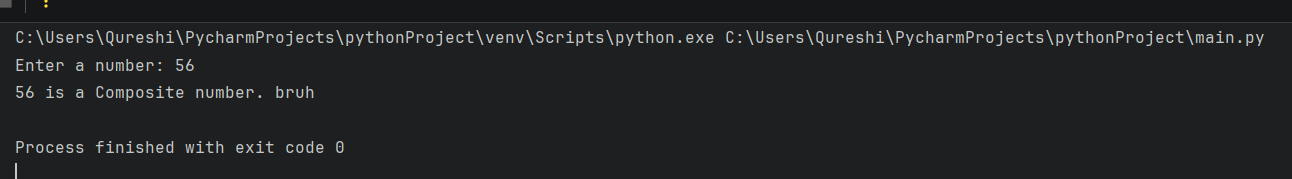
****

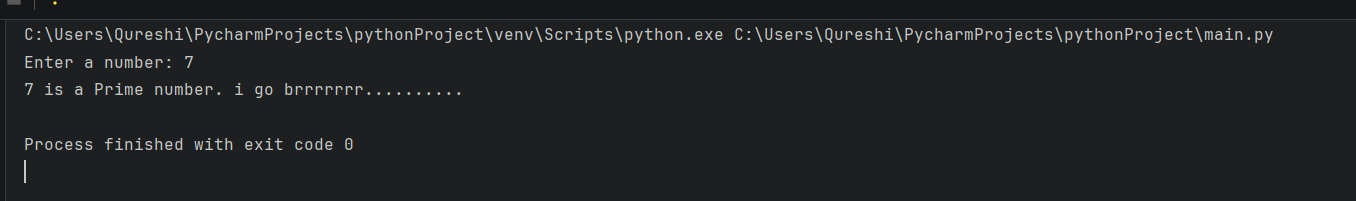
27) Check whether entered number is prime or not.

**CODE:**

n = int(input("Enter a number: "))  
isPrime = True  
for i in range(2, n):  
 if n % i == 0:  
 isPrime = False  
 break  
if isPrime:  
 print(n, "is a Prime number. i go brrrrrrr..........")  
else:  
 print(n, "is a Composite number. bruh")

**OUTPUT:**

****

****