**1.Write a program to calculate simple interest. formula to calculate simple interest : PNR/100**

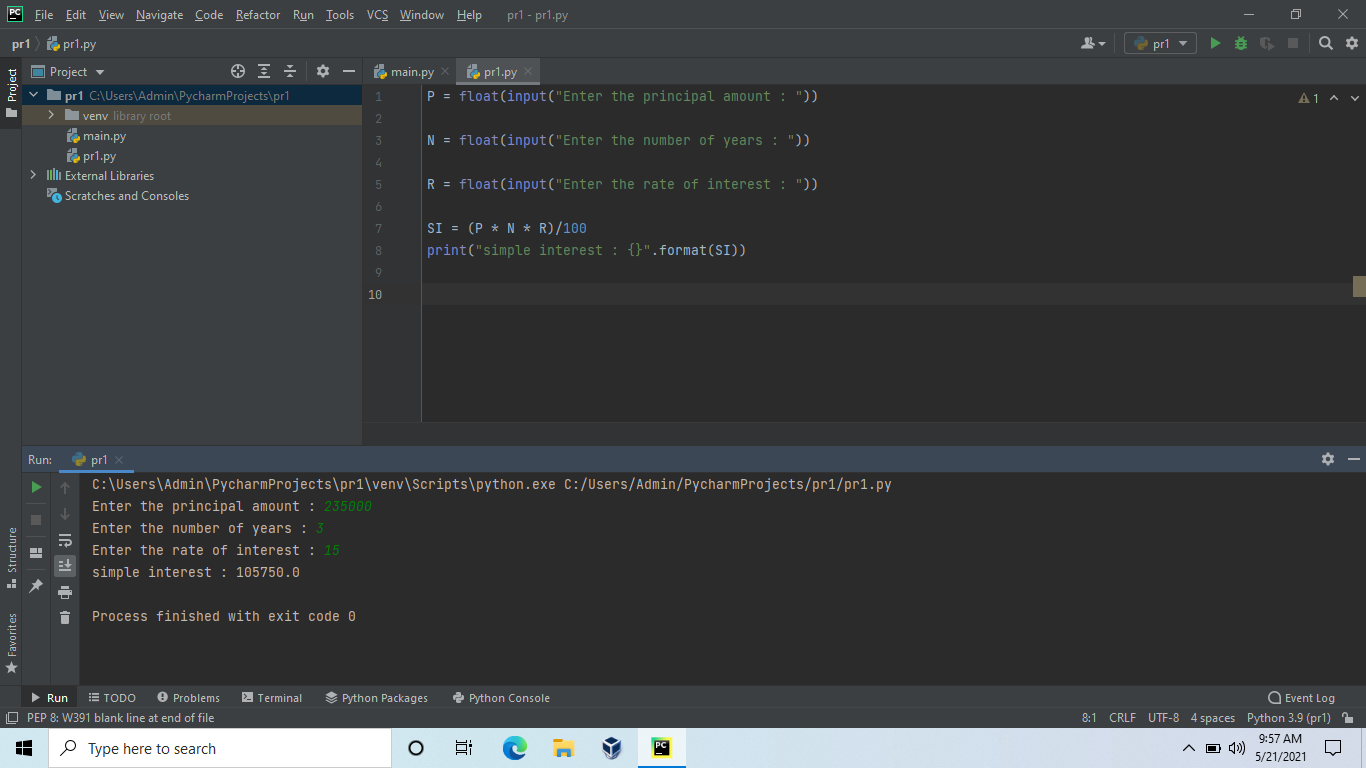
P = float(input("Enter the principal amount : "))

N = float(input("Enter the number of years : "))

R = float(input("Enter the rate of interest : "))

SI = (P \* N \* R)/100

print("simple interest : {}".format(SI))



**2. write a program to find sum and average of 3 values**

**a = int(input())**

**b = int(input())**

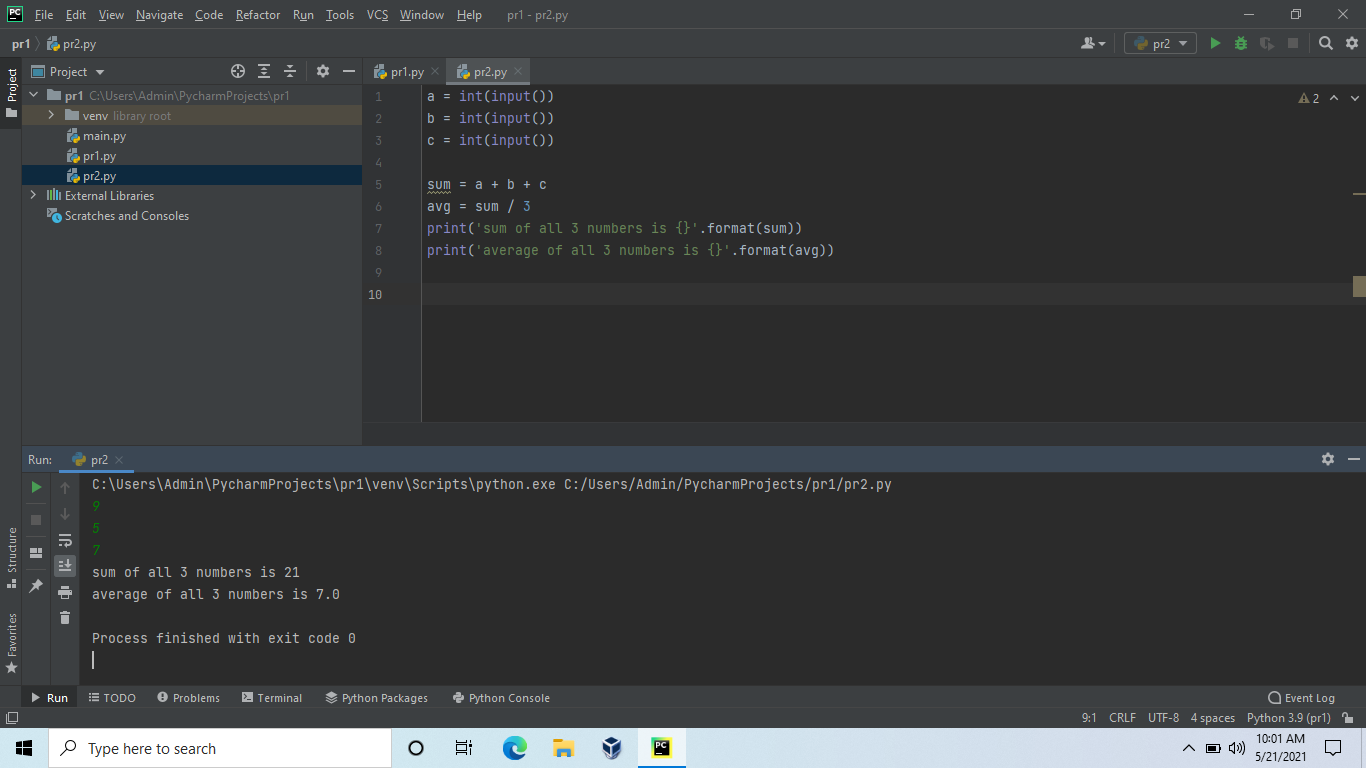
**c = int(input())**

**sum = a + b + c**

**avg = sum / 3**

**print('sum of all 3 numbers is {}'.format(sum))**

**print('average of all 3 numbers is {}'.format(avg))**

****

**3. Write a program to perform the following task.**

**The provided code stub reads two integers from STDIN,  and . Add code to print three lines where:**

**The first line contains the sum of the two numbers.**

**The second line contains the difference of the two numbers (first - second).**

**The third line contains the product of the two numbers.**

**a = int(input())**

**b = int(input())**

**if 1<=a<=10\*\*10 and 1<=b<=10\*\*10:**

**sum = a+b**

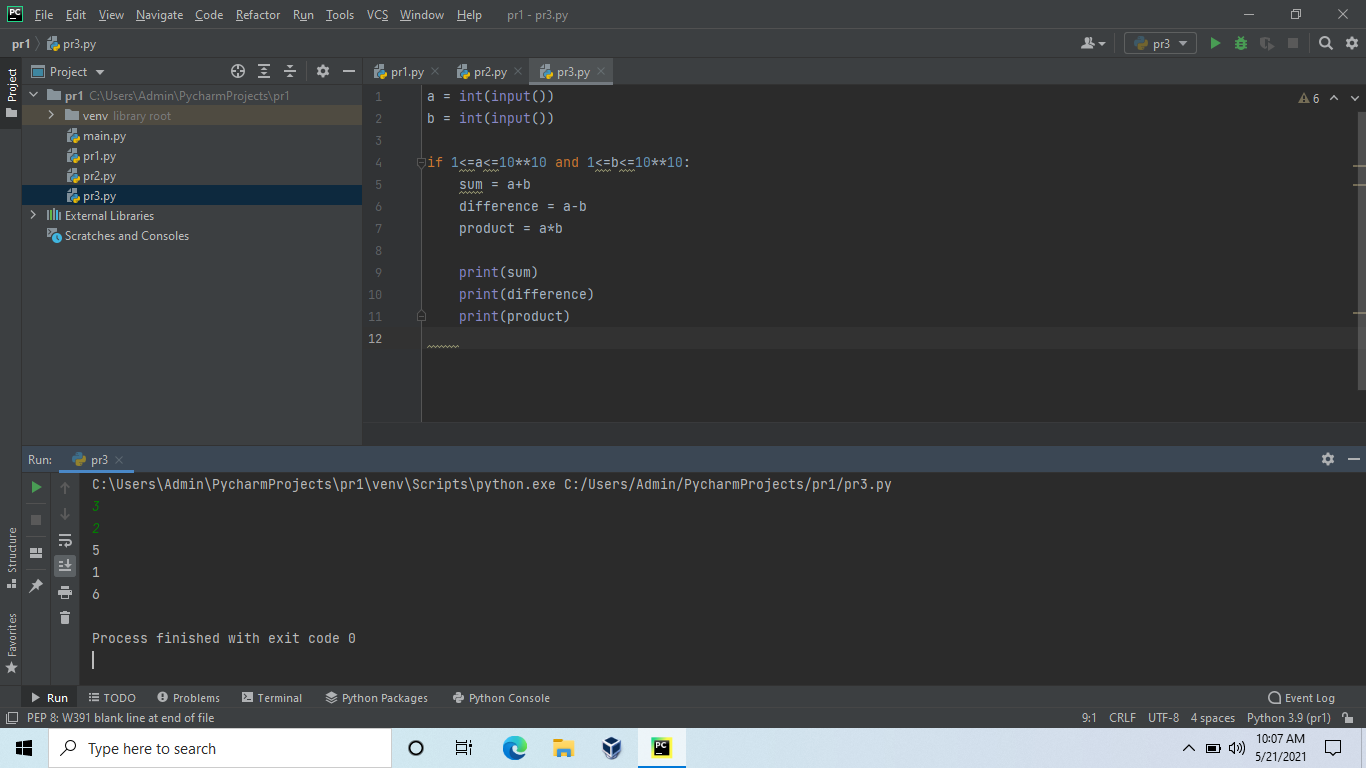
**difference = a-b**

**product = a\*b**

**print(sum)**

**print(difference)**

**print(product)**

****

**4.Read three integers from the keyboard a,b,c and those values in the following order.**

**max > mid > min**

**a = int(input())**

**b = int(input())**

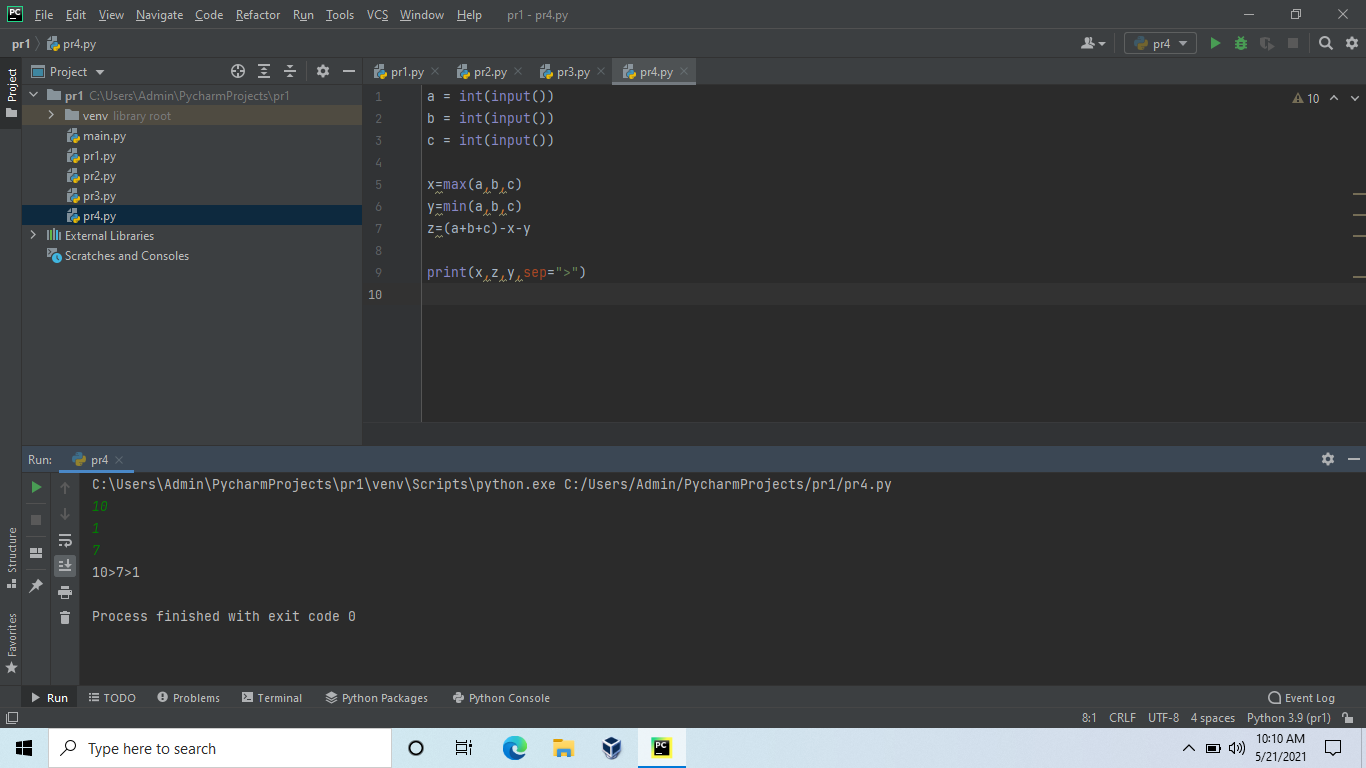
**c = int(input())**

**x=max(a,b,c)**

**y=min(a,b,c)**

**z=(a+b+c)-x-y**

**print(x,z,y,sep=">")**

****

**5.Read four values from the keyboard a,b,c,d and print the result of a^b + c^d in single line.**

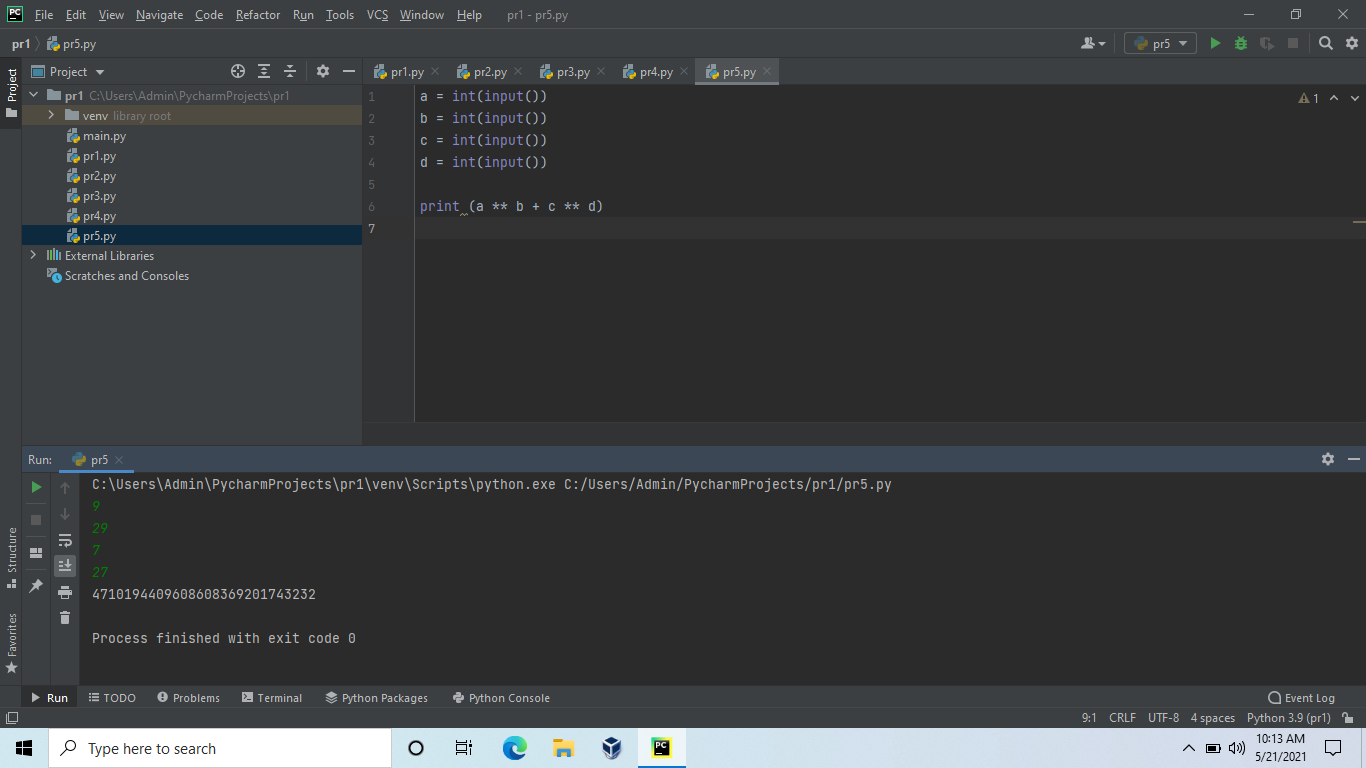
**a = int(input())**

**b = int(input())**

**c = int(input())**

**d = int(input())**

**print (a \*\* b + c \*\* d)**

****

**1.Given an integer,n, perform the following conditional actions:**

**If n is odd, print Weird**

**If n is even and**

**in the inclusive range of 2 to 5, print Not Weird**

**If n is even and in the inclusive range of 6 to 20, print     Weird**

**If n is even and greater than 20, print Not Weird**

**n = int(input())**

**if n % 2 == 1:**

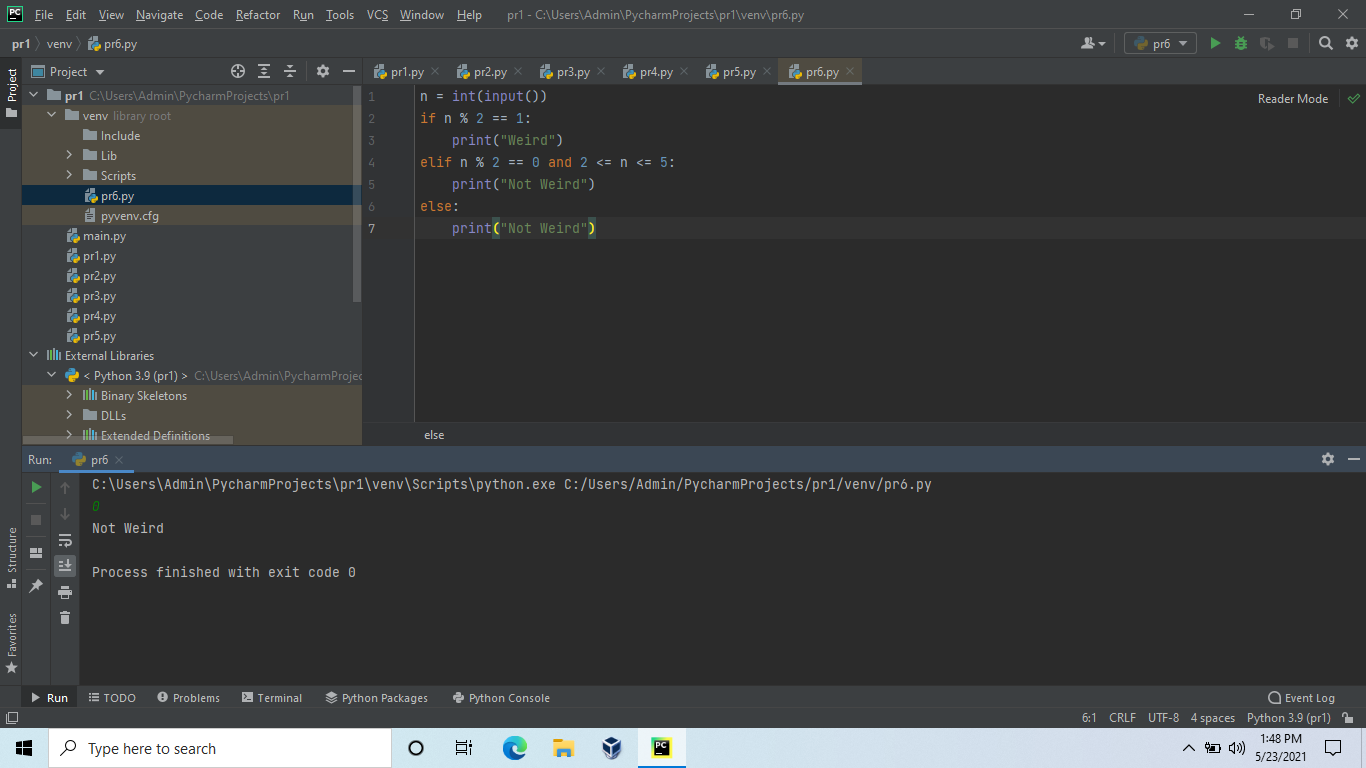
**print("Weird")**

**elif n % 2 == 0 and 2 <= n <= 5:**

**print("Not Weird")**

**else:**

**print("Not Weird")**

****

**2.Read three integers from the keyboard a,b,c, d and those values in the following order.**

**max > mid1 > mid2 > min**

**w = int(input())**

**x = int(input())**

**y = int(input())**

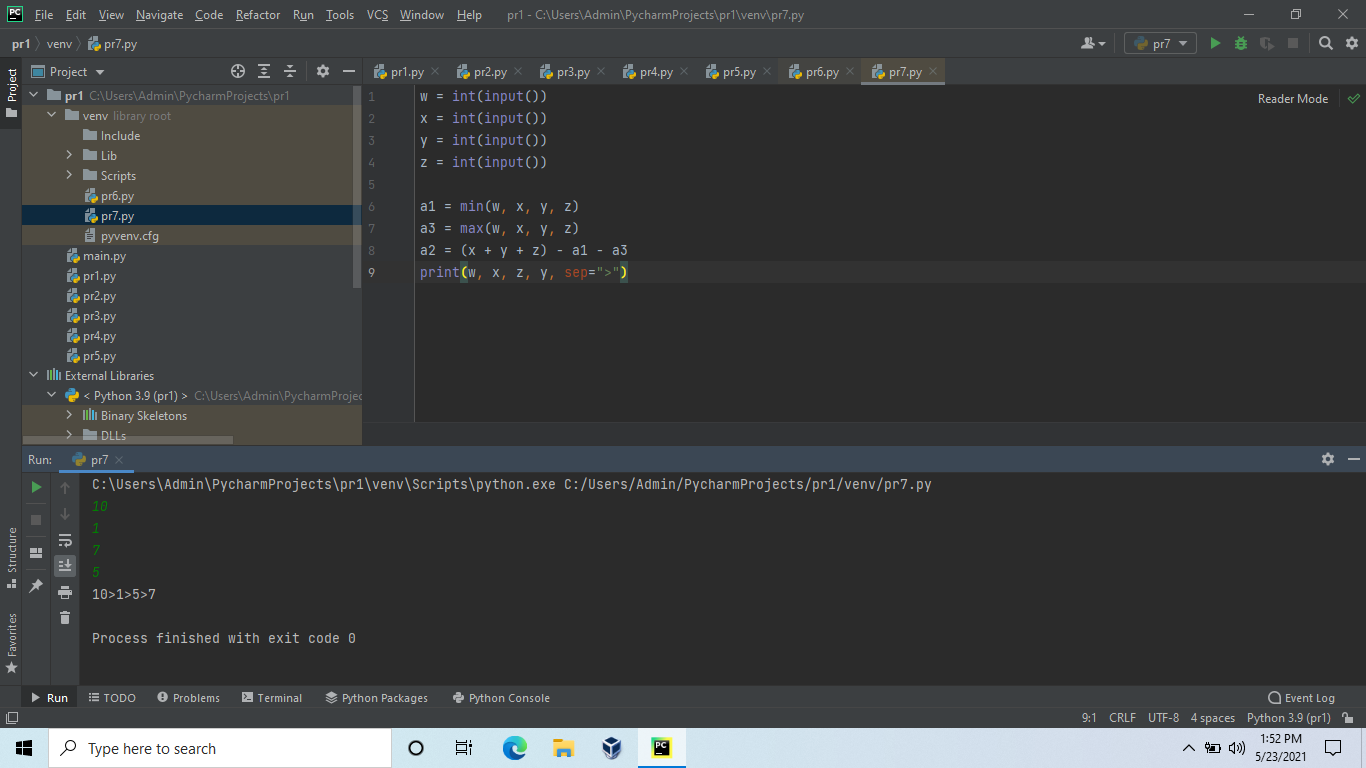
**z = int(input())**

**a1 = min(w, x, y, z)**

**a3 = max(w, x, y, z)**

**a2 = (x + y + z) - a1 - a3**

**print(w, x, z, y, sep=">")**

****

**3.Write a program to calculate the EB Bill.**

**The tariff rate for all division is the same. Karnataka electricity board single slaps for the domestic LT supply such as for 0 to 30 units the per-unit cost will be ? 3.75/-, from 31 to 100 the per-unit cost will be ? 5.20, from 101 to 200, the per-unit cost will be ? 6.75 and above 201 units you have to pay ? 7.8 per unit.**

**Additionally, the consumer will pay fixed charges as ? 60/- and electricity tax of 5% extra.**

**units = int(input(" Please enter the number of units you consumed : "))**

**if(units <= 30):**

**amount = units \* 3.75**

**surcharge = 60**

**elif(units <= 100):**

**amount = 130 + ((units - 30) \* 5.20)**

**surcharge = 70**

**elif(units <= 200):**

**amount = 130 + 162.50 + ((units - 100) \* 6.75)**

**surcharge = 80**

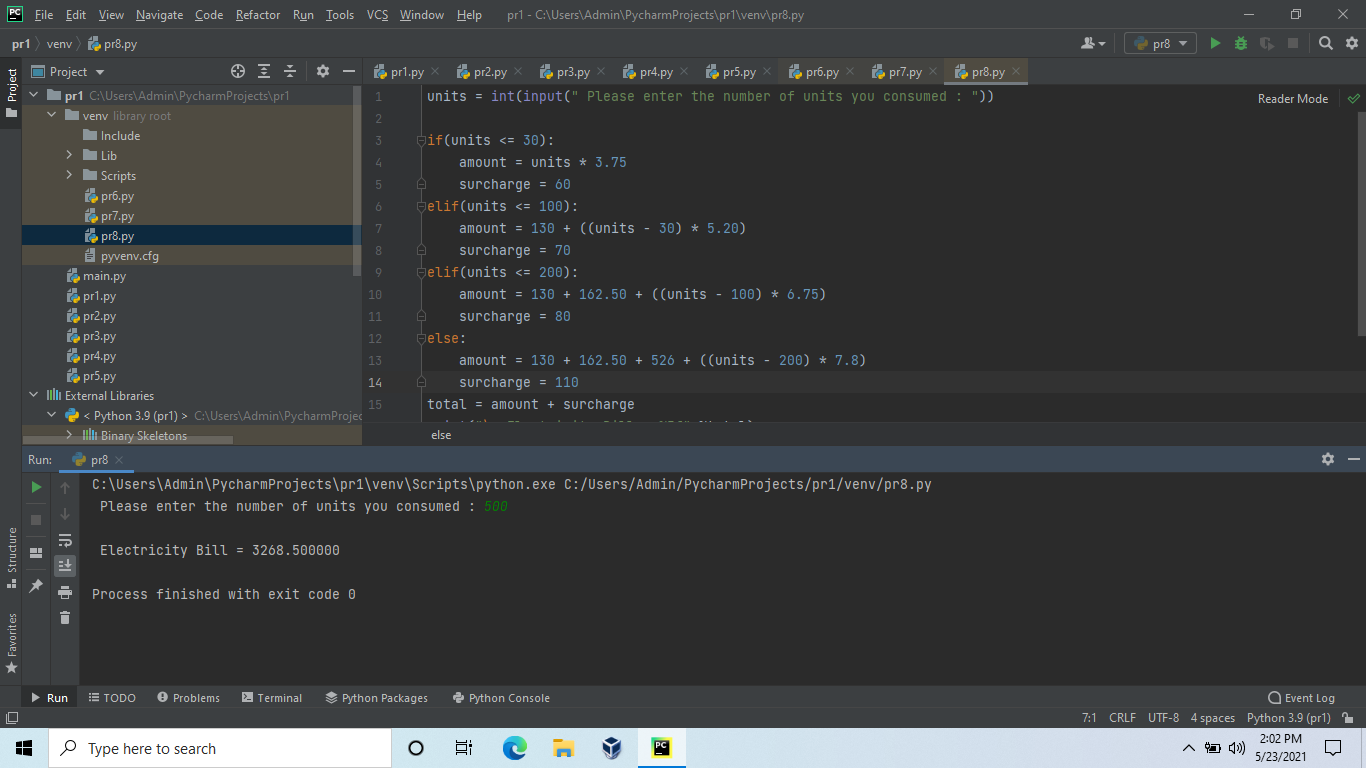
**else:**

**amount = 130 + 162.50 + 526 + ((units - 200) \* 7.8)**

**surcharge = 110**

**total = amount + surcharge**

**print("\n Electricity Bill = %5f" %total)**

****

**4.Write a program to calculate the grade. The grade should be         calculated in the following method.**

**Constraints**

**Score should be in between 1 to 100**

**Score**

**>= 90 --> Grade O**

**>= 80 --> Grade A+**

**>= 70 --> Grade A**

**>= 60 --> Grade B+**

**>= 50 --> Grade B**

**< 50 No Grade**

**markFour = int(input())**

**markFive = int(input())**

**tot = markOne + markTwo + markThree + markFour + markFive**

**avg = tot/5**

**if avg>=90:**

**print("Your grade is o")**

**elif avg>=80:**

**print("Your grade is A+")**

**elif avg>=70:**

**print("Your grade is A")**

**elif avg >= 60:**

**print("Your grade is B+")**

**elif avg>=50:**

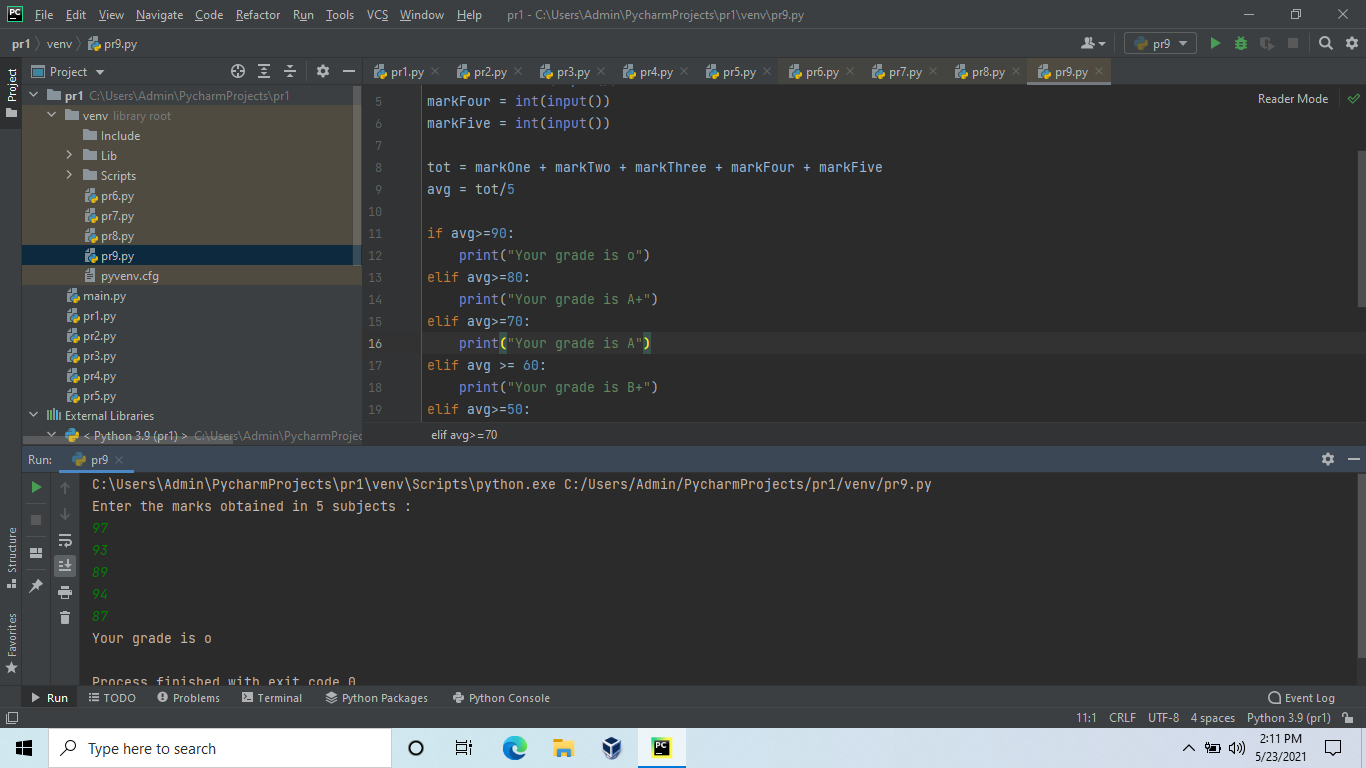
**print("Your grade is B")**

**elif avg<50:**

**print("No grade")**

**else:**

**print("invalid")**

****

**5.An extra day is added to the calendar almost every four years as February 29, and the day is called a leap day. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.**

**In the Gregorian calendar, three conditions are used to identify leap years:**

**The year can be evenly divided by 4, is a leap year, unless:**

**The year can be evenly divided by 100, it is NOT a leap year, unless:**

**The year is also evenly divisible by 400. Then it is a leap year.**

**This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years. Source**

**Task**

**Given a year, determine whether it is a leap year. If it is a leap year, print YES or NO.**

**year = int(input("Enter a year: "))**

**if (year % 4) == 0:**

**if (year % 100) == 0:**

**if (year % 400) == 0:**

**print("YES".format(year))**

**else:**

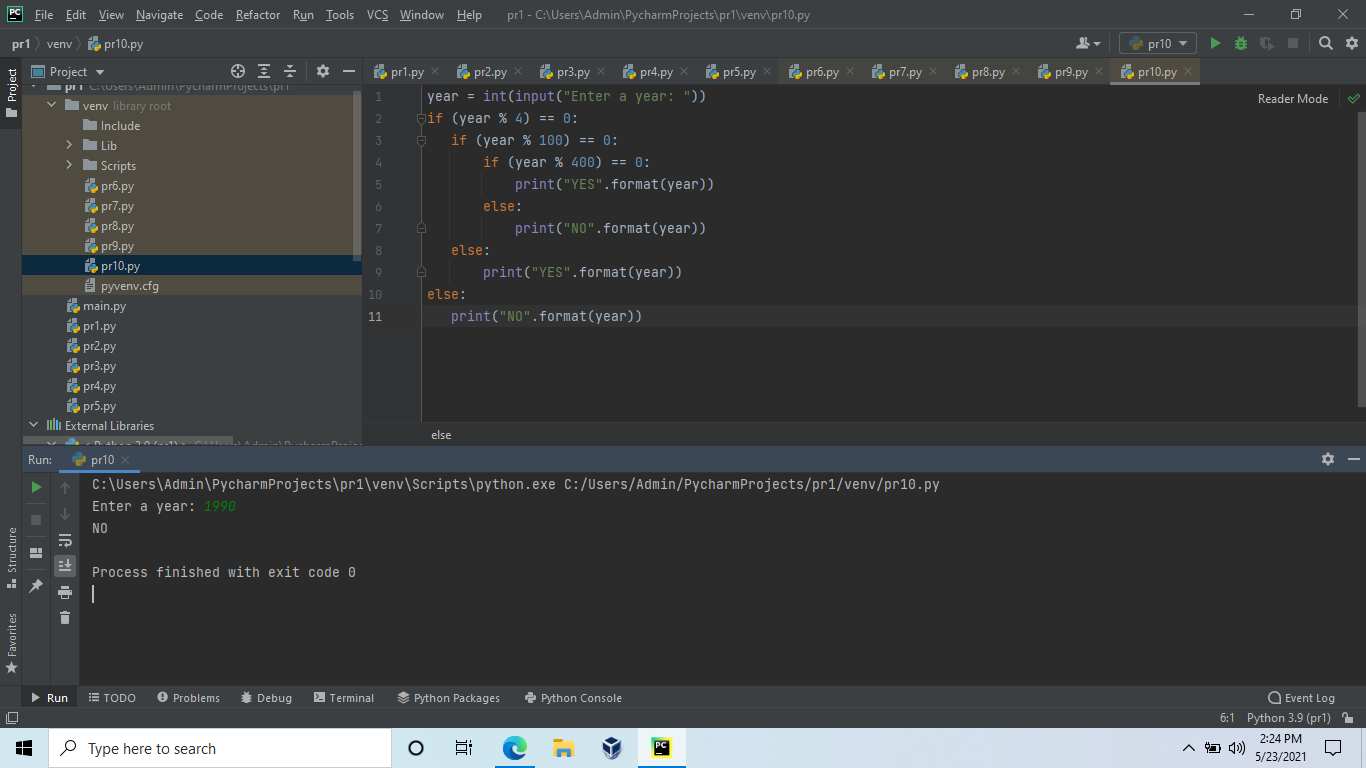
**print("NO".format(year))**

**else:**

**print("YES".format(year))**

**else:**

**print("NO".format(year))**

****

**1.A little girl living in a village craves some rava idli even though she has had rava idli for the last 346514534 days in a row !! (Strange, you might think. But its normal down here)**

**At the idli shop there are two types of Rava Idli's available.**

**One goes for Rs.A per piece and the other goes for Rs.B per piece.**

**The girl has a total of K rupees.**

**What is the maximum number of rava idlis that she can have?**

**Note that she does not care about the type of idli she gets, she just wants to have as many rava idlis of any type as possible.**

**Input**

**The first line contains the number of test cases T**

**1 = T = 1000**

**Each test case contains three integers, A, B and K.**

**1 = A,B,K = 10^9**

**Output**

**Print the maximum number of idlis she can buy for each test case on a new line**

**n=int(input())**

**for i in range (n):**

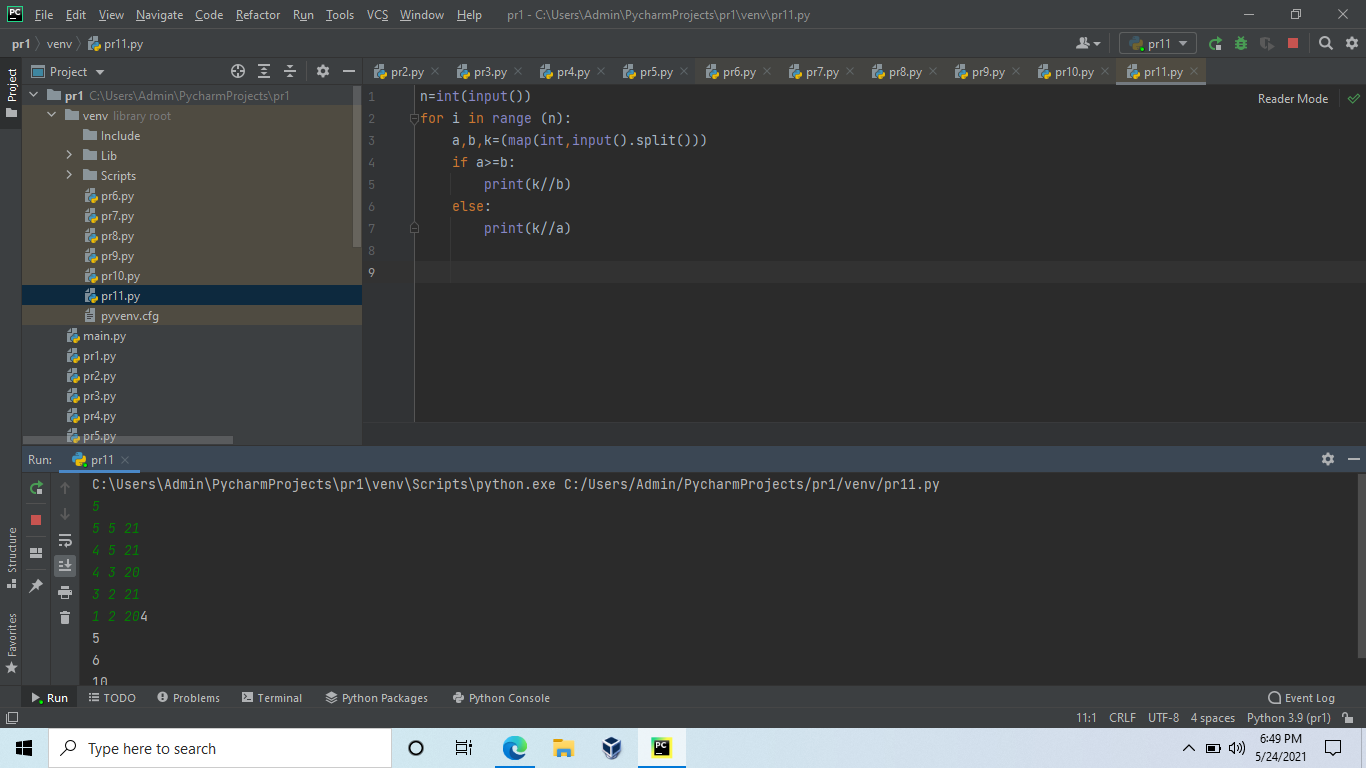
**a,b,k=(map(int,input().split()))**

**if a>=b:**

**print(k//b)**

**else:**

**print(k//a)**

****

**2.You are given a positive integer . Print a numerical triangle of height  like the one below:**

**1**

**22**

**333**

**4444**

**55555**

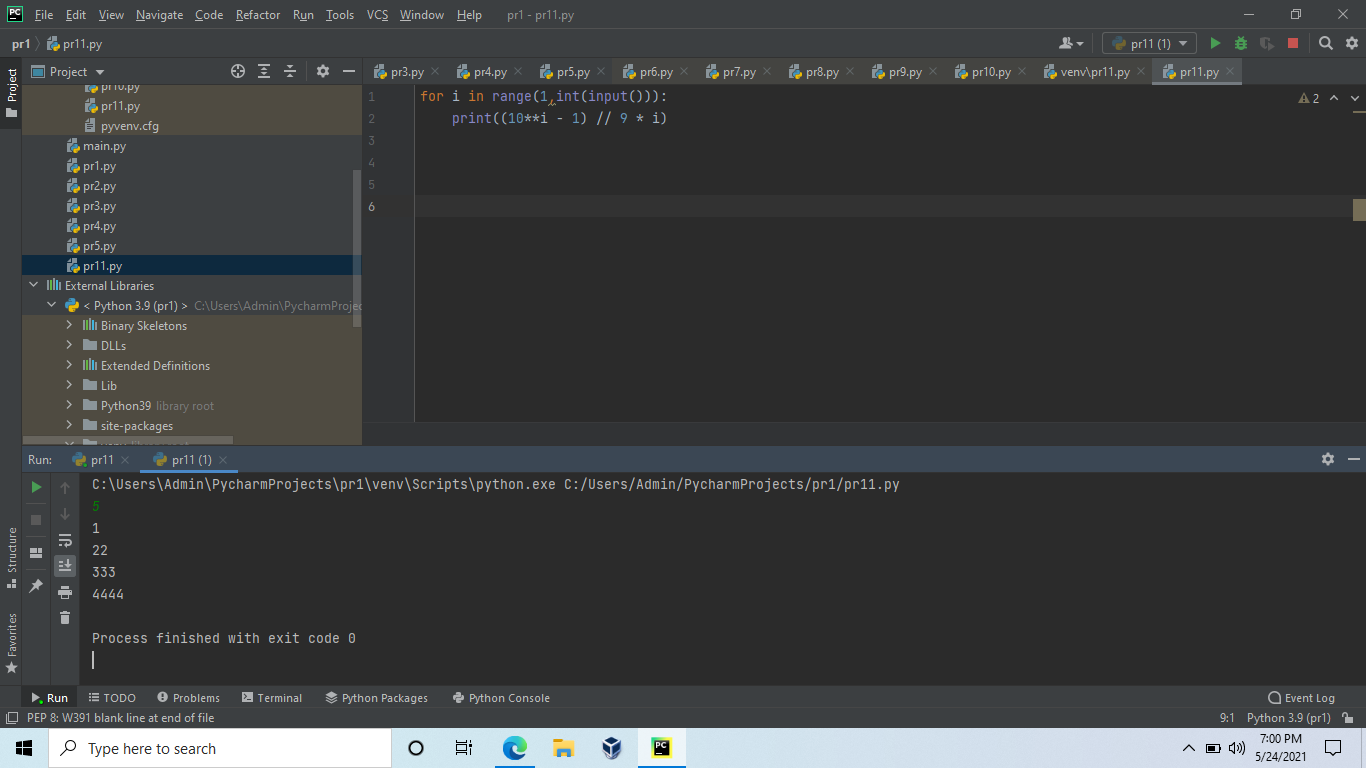
**......**

**Can you do it using only arithmetic operations, a single for loop and print statement?**

**Use no more than two lines. The first line (the for statement) is already written for you. You have to complete the print statement.**

**for i in range(1,int(input())):**

**print((10\*\*i - 1) // 9 \* i)**

****

**3.You are given a rectangular board of M×N squares. Also you are given an unlimited number of standard domino pieces of 2×1 squares. You are allowed to rotate the pieces. You are asked to place as many dominoes as possible on the board so as to meet the following conditions:**

**1.Each domino completely covers two squares.**

**2.No two dominoes overlap.**

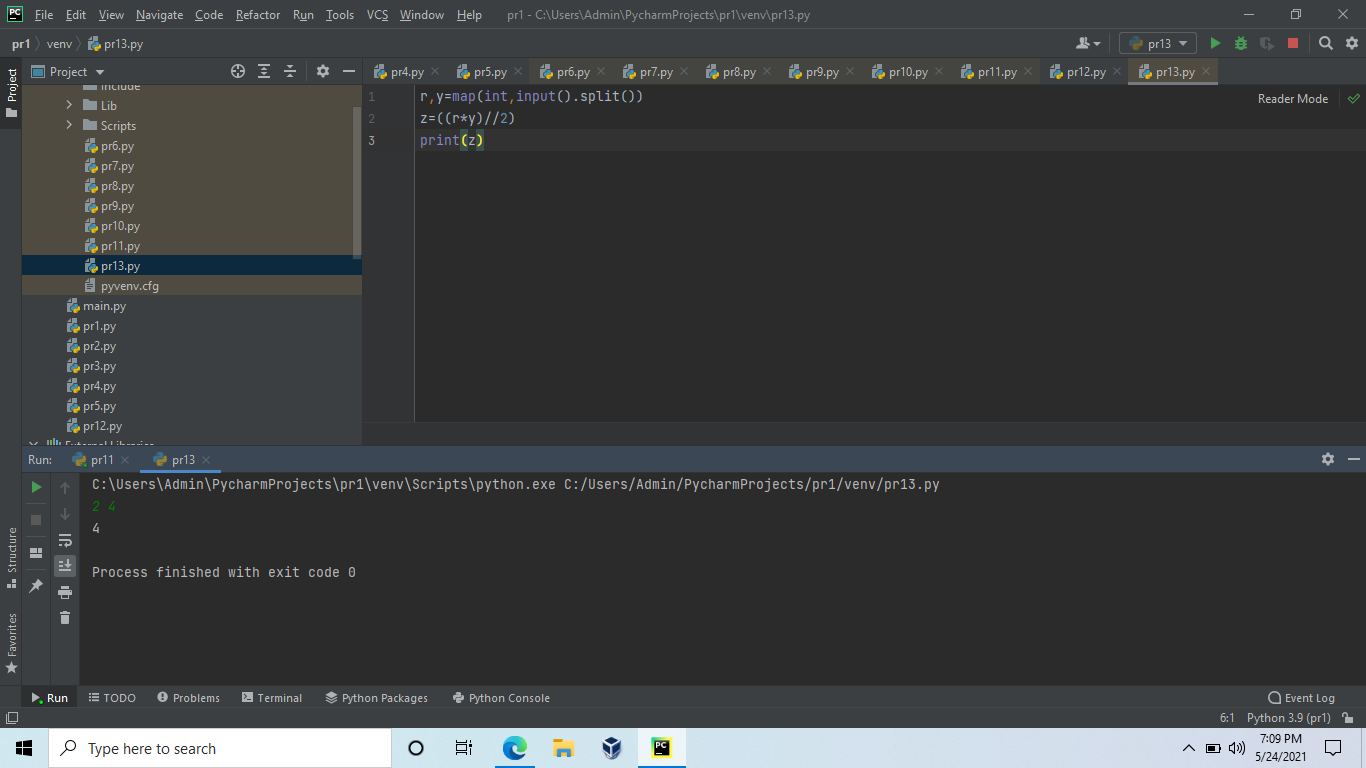
**3.Each domino lies entirely inside the board. It is allowed to touch the edges of the board.**

**Find the maximum number of dominoes, which can be placed under these restrictions.**

**r,y=map(int,input().split())**

**z=((r\*y)//2)**

**print(z)**

****

**4.Given an integer,n, perform the following conditional actions:**

**If n is odd, print Weird If n is even and in the inclusive range of 2 to 5, print Not Weird If n is even and in the inclusive range of 6 to 20, print Weird If n is even and greater than 20, print Not Weird.**

import math

import os

import random

import re

import sys

n = int(input())

if n % 2 != 0:

   print("Weird")

elif 2 <= n <= 5:

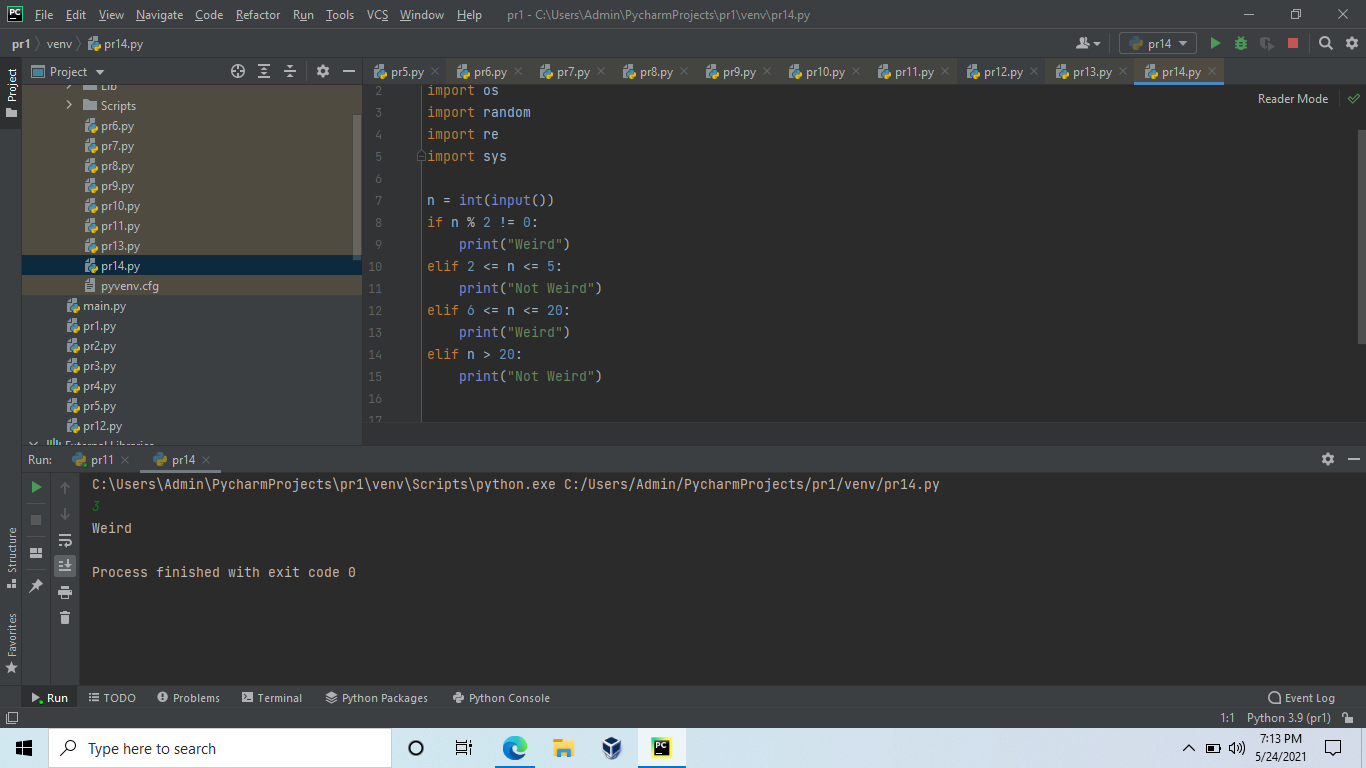
   print("Not Weird")

elif 6 <= n <= 20:

   print("Weird")

elif n > 20:

   print("Not Weird")



**5.PRINT THE BELOW MENTIONED PATTERN FOR ANY "N" VALUE. WHERE "N" INDICATES NO.OF ROWS.**

**Input Format**

**A SINGLE INTEGER DENOTING N VALUE**

**Constraints**

**1<=N<=100**

**N is only odd**

**Output Format**

**PATTERN AS SHOWN IN SAMPLE TEST CASE.**

n = int(input())

for i in range(1, n // 2 + 2):

   for j in range(1, n + 1):

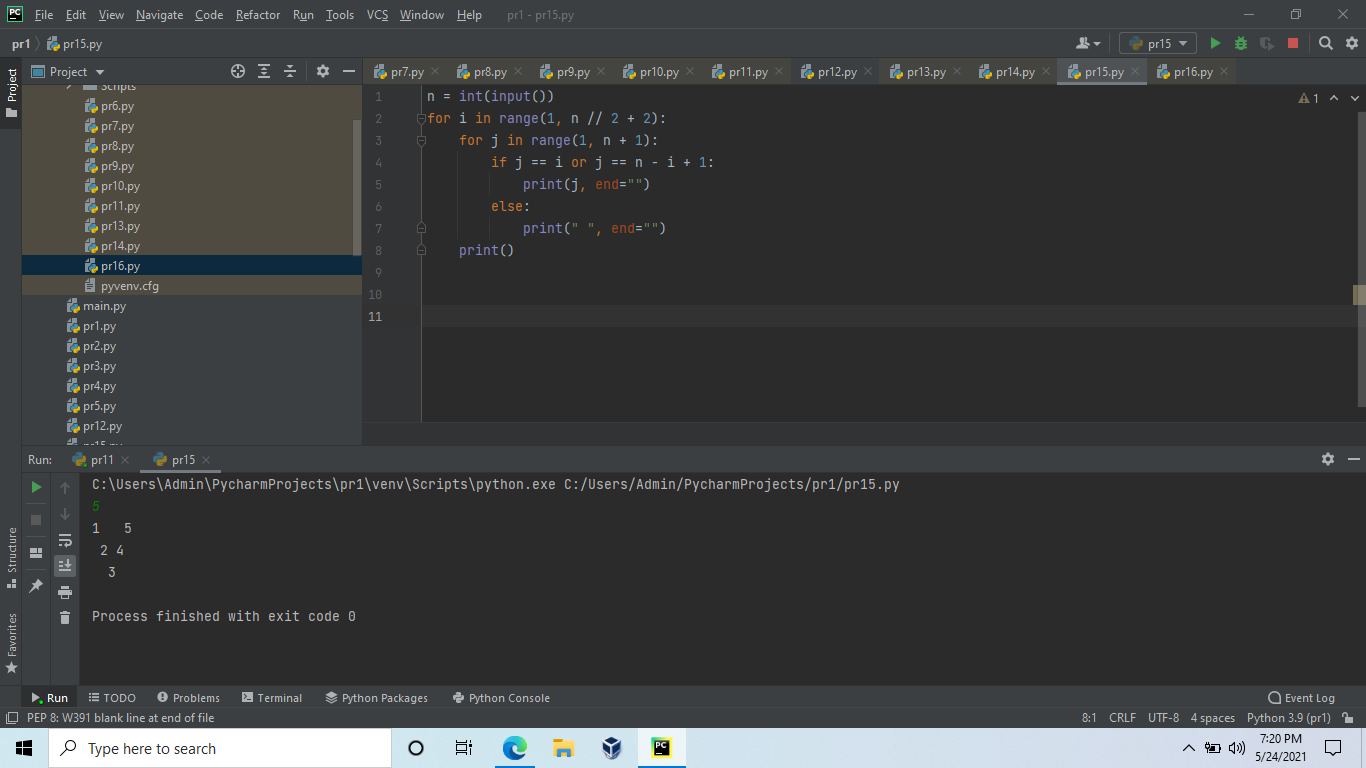
       if j == i or j == n - i + 1:

           print(j, end="")

       else:

           print(" ", end="")

   print()



**6.PRINT THE BELOW MENTIONED PATTERN FOR ANY "N" VALUE. WHERE "N" INDICATES NO.OF ROWS.**

**Input Format**

**A SINGLE INTEGER DENOTING N VALUE**

**Constraints**

**1<=N<=100**

**Output Format**

**PATTERN AS SHOWN IN SAMPLE TEST CASE.**

n = int(input())

for i in range(1, n + 1):

   for j in range(1, n + 1):

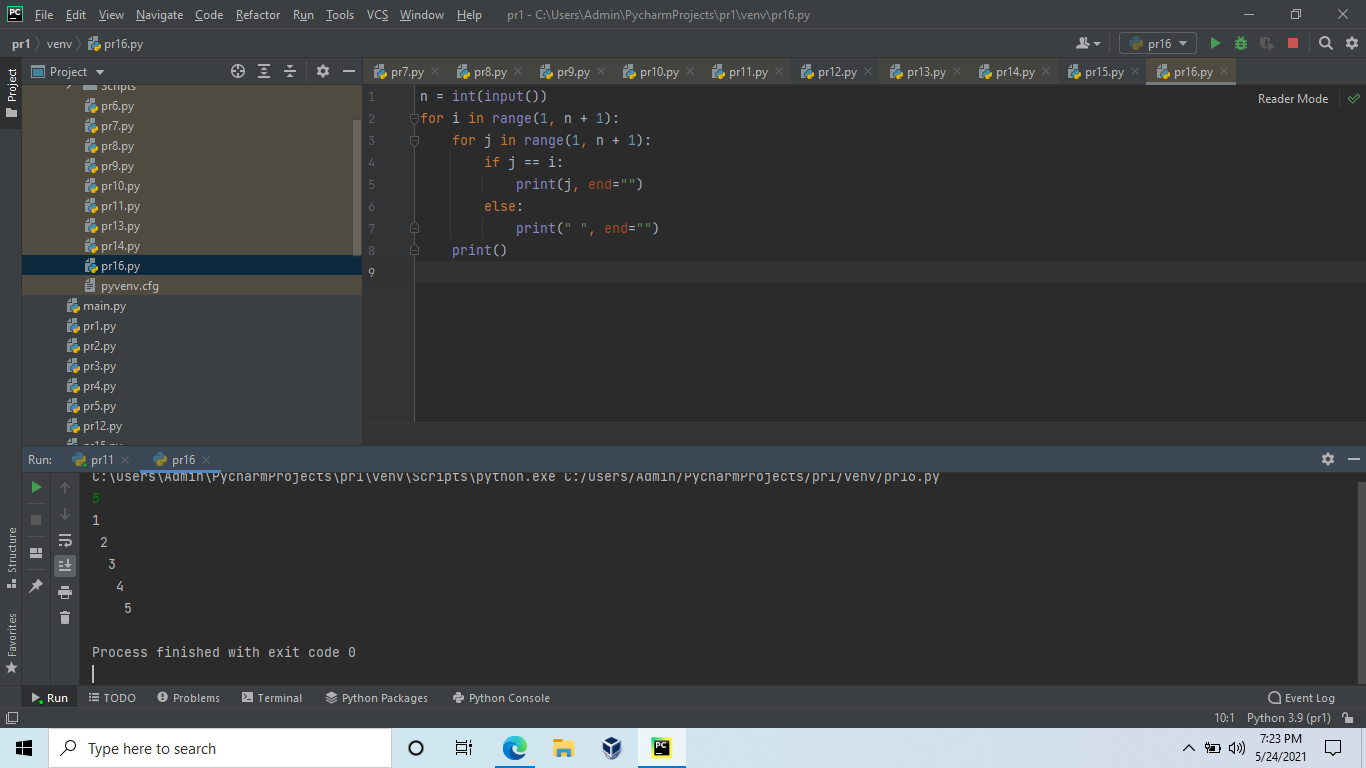
       if j == i:

           print(j, end="")

       else:

           print(" ", end="")

   print()



**1.Given the participants' score sheet for your University Sports Day, you are required to find the runner-up score. You are given N scores. Store them in a list and find the score of the runner-up.**

**if \_\_name\_\_=="\_\_main\_\_":**

**n = int(input())**

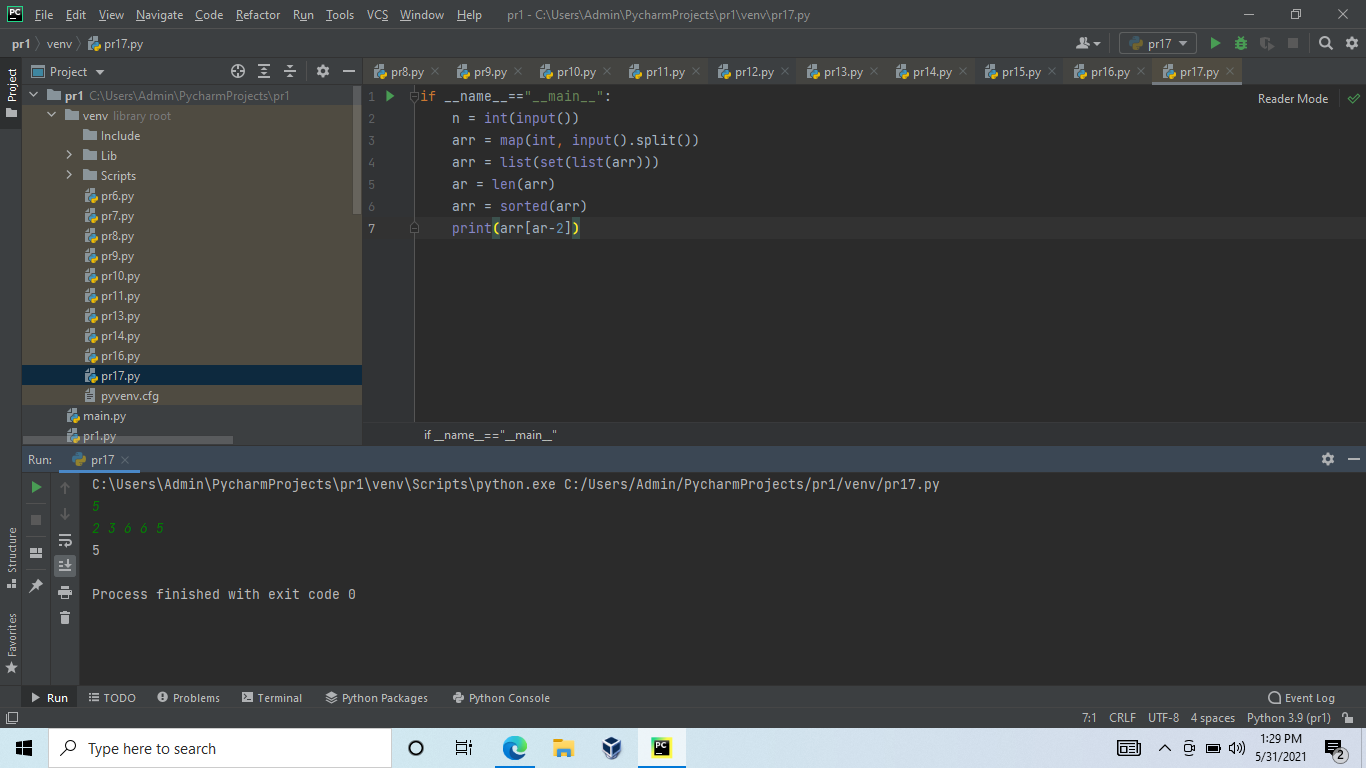
**arr = map(int, input().split())**

**arr = list(set(list(arr)))**

**ar = len(arr)**

**arr = sorted(arr)**

**print(arr[ar-2])**

****

**2.Rotate a given String in the specified direction by specified magnitude.**

**After each rotation make a note of the first character of the rotated String, After all rotation are performed the accumulated first character as noted previously will form another string, say FIRSTCHARSTRING.**

**Check If FIRSTCHARSTRING is an Anagram of any substring of the Original string.**

**If yes print "YES" otherwise "NO". Input**

**The first line contains the original string s. The second line contains a single integer q. The ith of the next q lines contains character d[i] denoting direction and integer r[i] denoting the magnitude.**

**Constraints**

**1 <= Length of original string <= 30**

**1<= q <= 10**

**Output**

**YES or NO**

**from collections import Counter**

**def lrotate(input,d):**

**Lfirst = input[0 : d]**

**Lsecond = input[d :]**

**return (Lsecond + Lfirst)**

**def rrotate(input,d):**

**Rfirst = input[0 : len(input)-d]**

**Rsecond = input[len(input)-d : ]**

**return (Rsecond + Rfirst)**

**s=input()**

**n=int(input())**

**c='FIRSTCHARSTRING'**

**l=[]**

**for \_ in range(n):**

**w,v=input().split()**

**v=int(v)**

**if w == 'L':**

**p=lrotate(c,v)**

**if w == 'R':**

**p=rrotate(c,v)**

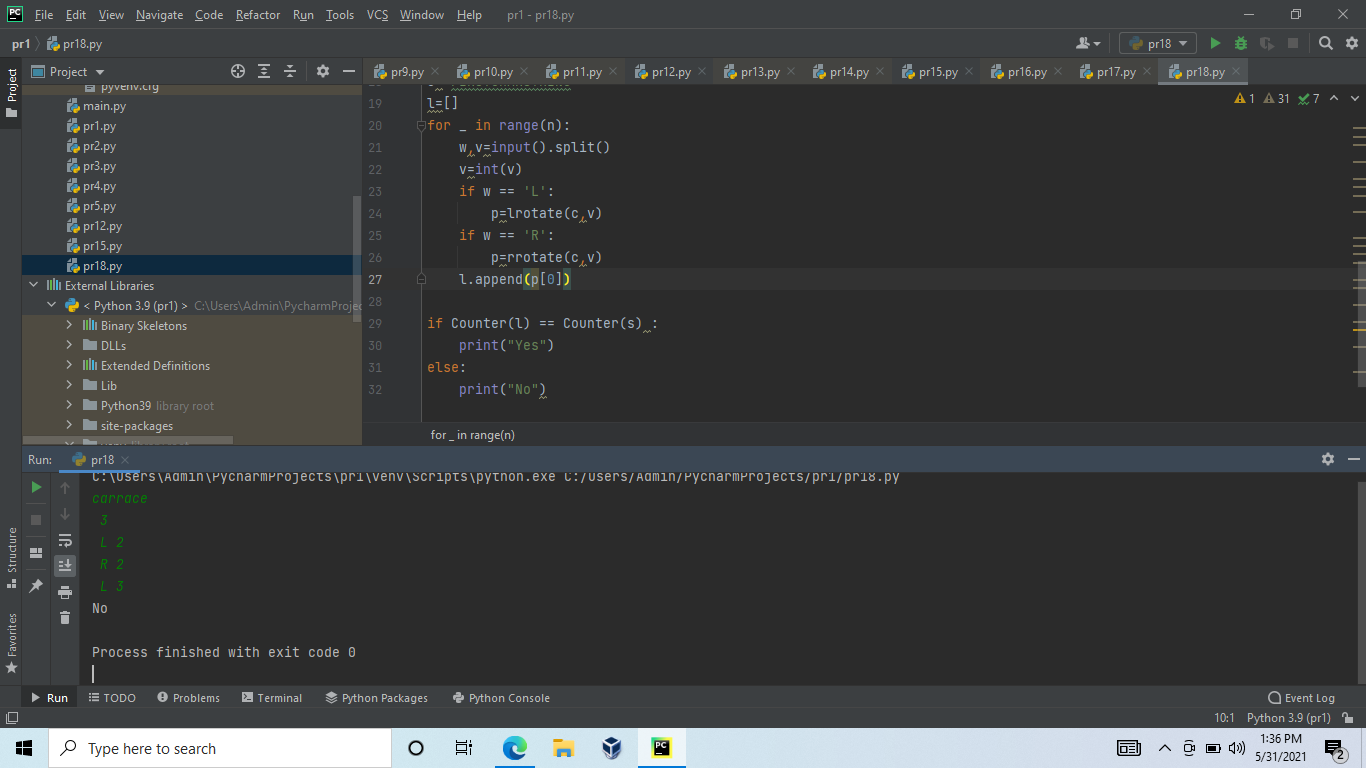
**l.append(p[0])**

**if Counter(l) == Counter(s) :**

**print("Yes")**

**else:**

**print("No")**

****

**3.Stubborn Vowels**

**In english alphabets there are two types of words, vowels and consonents.You are writing a program to reverse a given string, but the vowels are stubborn to move away from their position. So given a string where the vowels are stubborn,**

**print what will be word if the entire word is reversed except for the vowels.**

**Input Format**

**One string input**

**Constraints**

**3<=String length<=10^5**

**Output Format**

**Reversed string output**

**import re**

**string = input()**

**temp1 = string[:]**

**v = re.split('b|c|d|f|g|h|j|k|l|m|n|p|q|r|s|t|v|w|x|y|z', temp1)**

**c = re.split('a|e|i|o|u', string)**

**flag = 0**

**if v[0] == '':**

**flag = 1**

**c.reverse()**

**while '' in v:**

**v.remove('')**

**while '' in c:**

**c.remove('')**

**val = []**

**if flag == 1:**

**for i in range(len(c)):**

**val.append(c[i])**

**if (i < len(v)):**

**val.append(v[i])**

**else:**

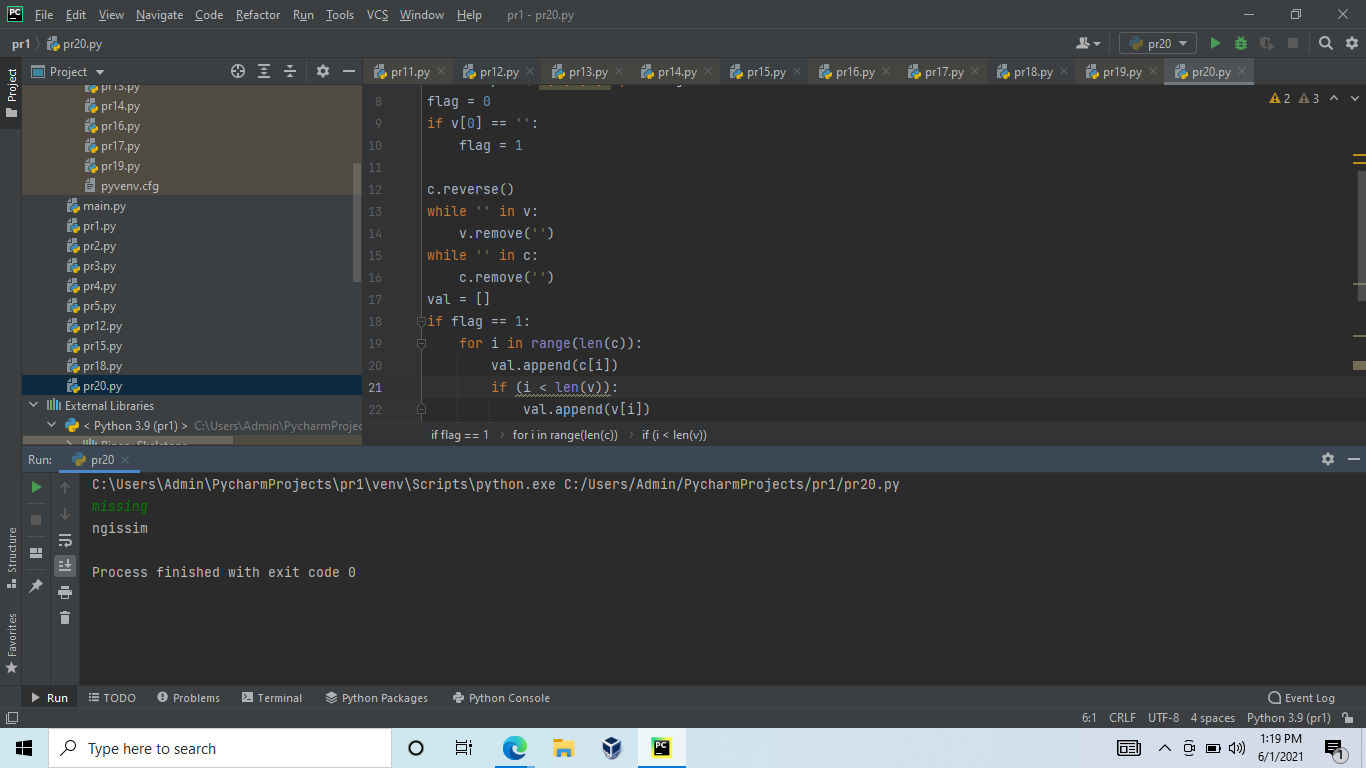
**for i in range(len(v)):**

**val.append(v[i])**

**if (i < len(c)):**

**val.append(c[i])**

**print("".join(val))**

****

**4.Array of integers is a hill, if:**

**it is strictly increasing in the beginning; after that it is constant; after that it is strictly decreasing.**

**The first block (increasing) and the last block (decreasing) may be absent. It is allowed that both of this blocks are absent.**

**For example, the following three arrays are a hill: [5,7,11,11,2,1], [4,4,2], [7],**

**but the following three are not unimodal: [5,5,6,6,1], [1,2,1,2], [4,5,5,6].**

**Write a program that checks if an array is a hill.**

**a, lst = int(input()), input().split()**

**lst = [int(ch) for ch in lst]**

**def f(m, n, flag):**

**if flag == 1:**

**return m < n**

**elif flag == 2:**

**return m == n**

**else:**

**return m > n**

**def is\_hill(a, lst):**

**flag = 1**

**for i in range(1, a):**

**if f(lst[i - 1], lst[i], flag):**

**continue**

**else:**

**if flag == 3:**

**return "NO"**

**flag += 1**

**if f(lst[i - 1], lst[i], flag):**

**continue**

**else:**

**if flag == 3:**

**return "NO"**

**flag += 1**

**if f(lst[i - 1], lst[i], flag):**

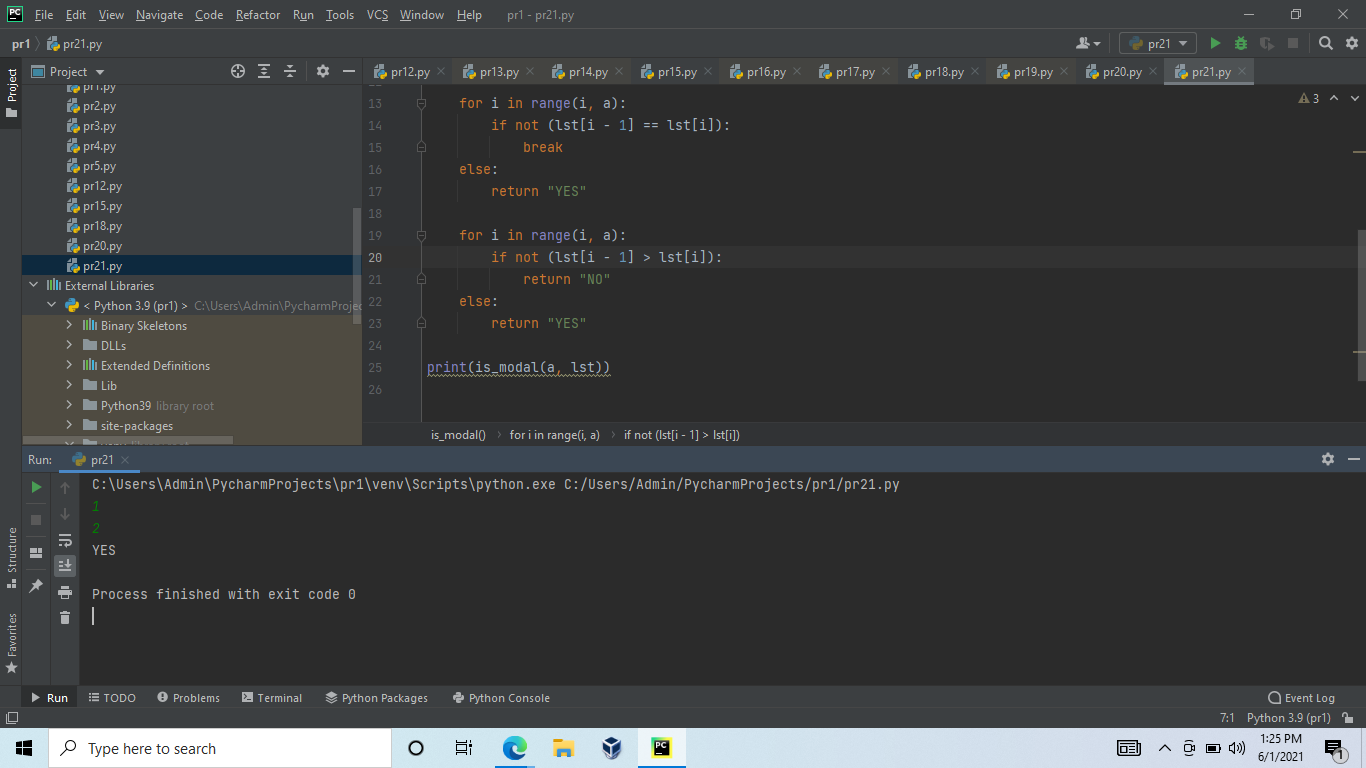
**continue**

**else:**

**return "NO"**

**return "YES"**

**print(is\_hill(a,lst)**

****

**5.A little girl living in a village craves some rava idli even though she has had rava idli for the last 346514534 days in a row !! (Strange, you might think. But its normal down here)**

**At the idli shop there are two types of Rava Idli's available.**

**One goes for Rs.A per piece and the other goes for Rs.B per piece.**

**The girl has a total of K rupees.**

**What is the maximum number of rava idlis that she can have?**

**Note that she does not care about the type of idli she gets, she just wants to have as many rava idlis of any type as possible.**

**Input**

**The first line contains the number of test cases T**

**1 = T = 1000**

**Each test case contains three integers, A, B and K.**

**1 = A,B,K = 10^9**

**Output**

**Print the maximum number of idlis she can buy for each test case on a new line.**

n=int(input())

for i in range (n):

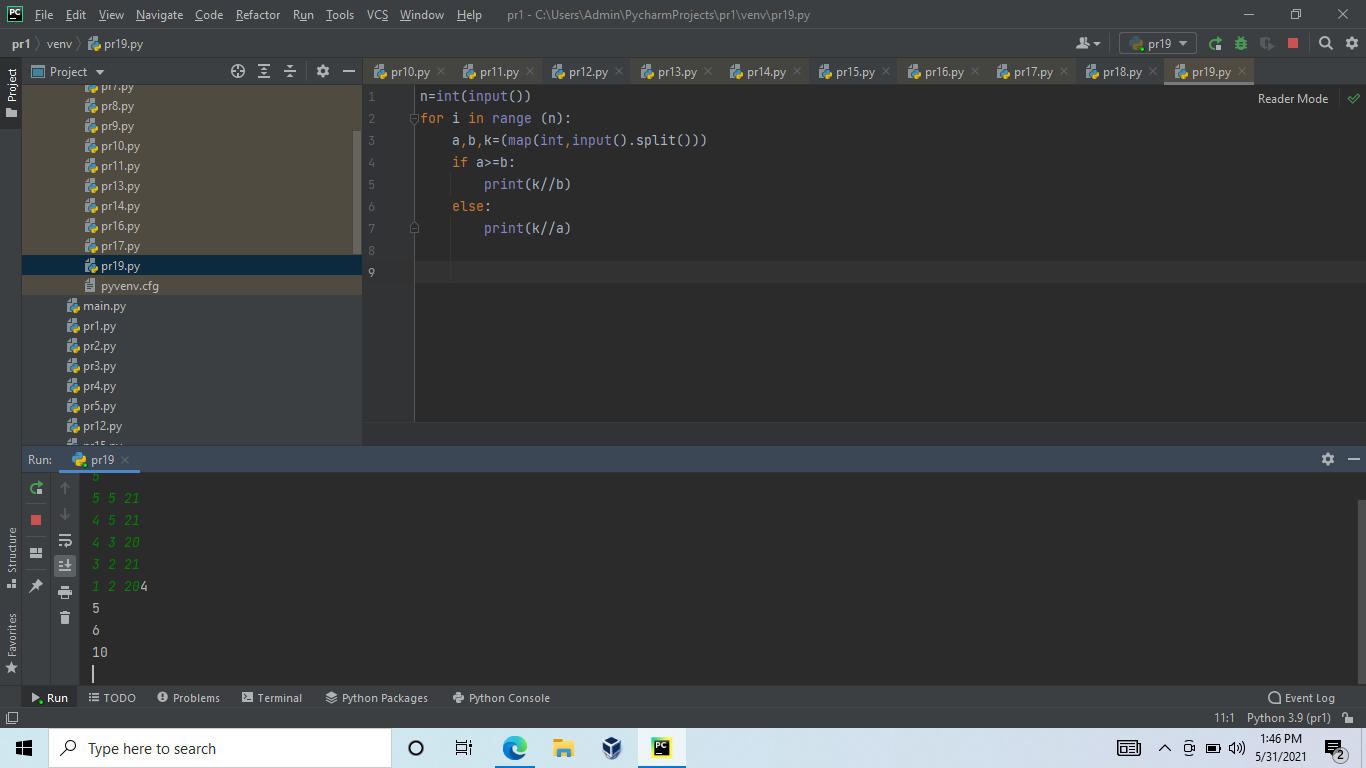
   a,b,k=(map(int,input().split()))

   if a>=b:

       print(k//b)

   else:

       print(k//a)



**6.Python program to remove Nth occurrence of the given word. Given a list of words in Python, the task is to remove the Nth occurrence of the given word in that list.**

**def RemoveIthWord(lst, word, N):**

**newList = []**

**count = 0**

**# iterate the elements**

**for i in lst:**

**if (i == word):**

**count = count + 1**

**if (count != N):**

**newList.append(i)**

**else:**

**newList.append(i)**

**lst = newList**

**if count == 0:**

**print("Item not found")**

**else:**

**print("Updated list is: ", lst)**

**return newList**

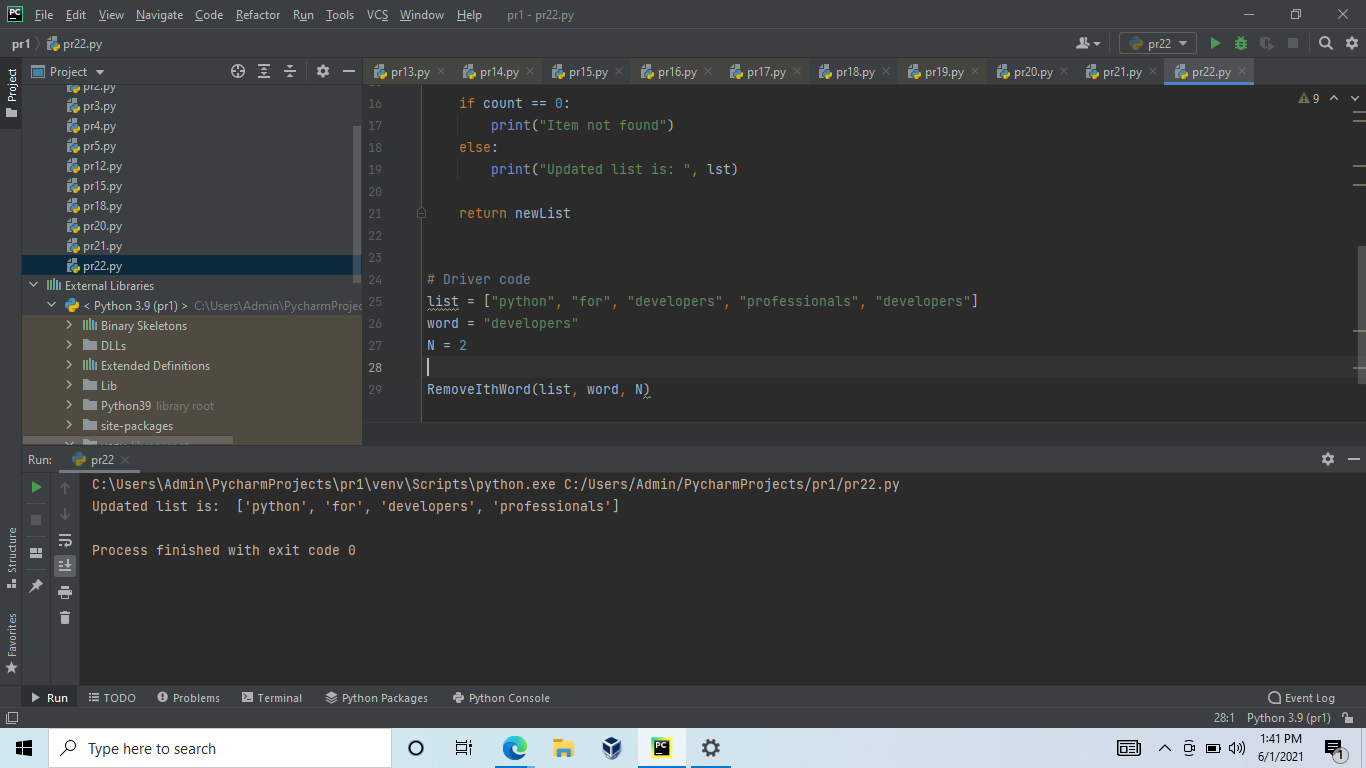
**# Driver code**

**list = ["python", "for", "developers", "professionals", "developers"]**

**word = "developers"**

**N = 2**

**RemoveIthWord(list, word, N)**

****

**7.Reverse words in a given String in Python**

**We are given a string and we need to reverse words of given string ?**

**def rev\_sentence(sentence):**

**# first split the string into words**

**words = sentence.split(' ')**

**# then reverse the split string list and join using space**

**reverse\_sentence = ' '.join(reversed(words))**

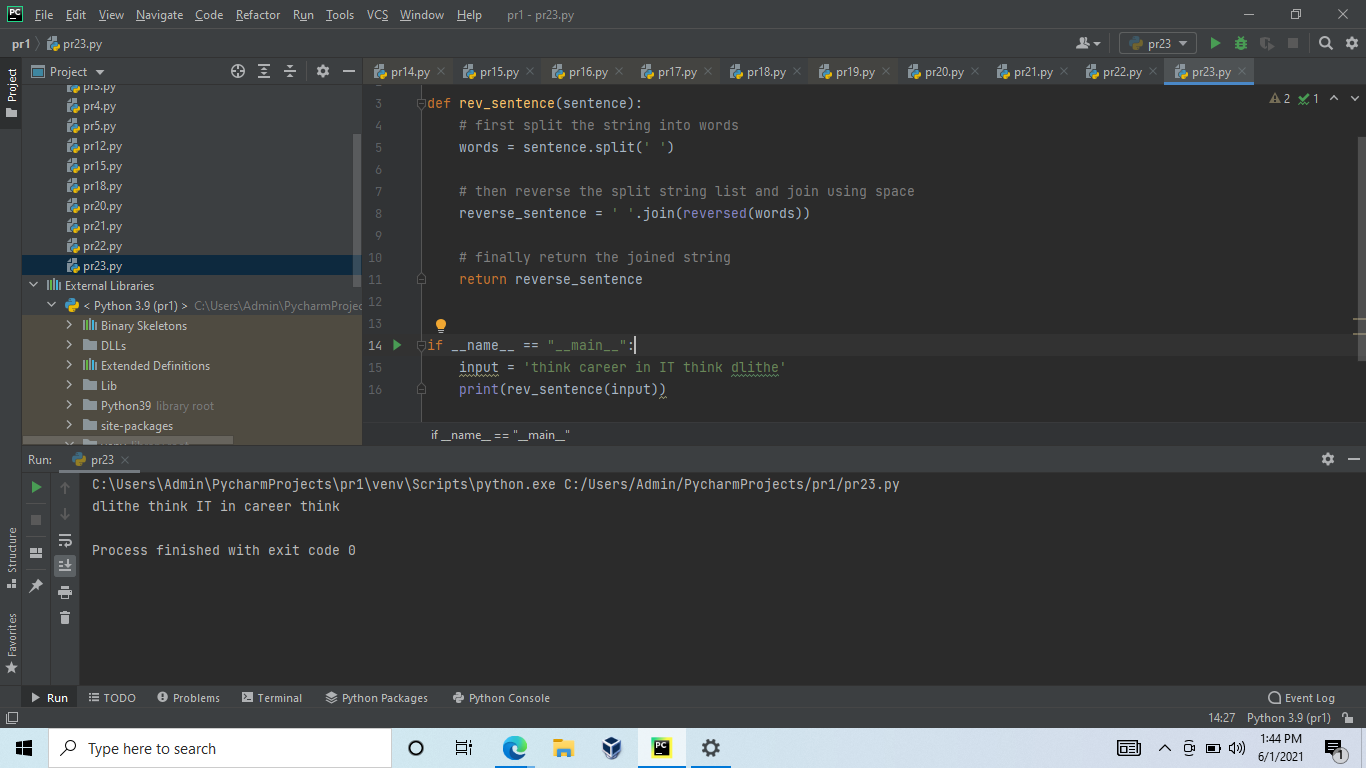
**# finally return the joined string**

**return reverse\_sentence**

**if \_\_name\_\_ == "\_\_main\_\_":**

**input = 'think career in IT think dlithe'**

**print(rev\_sentence(input))**

****

**8.Check if a Substring is Present in a Given String**

**Given two strings, check if s1 is there in s2.**

**# there in big string**

**def check(string, sub\_str):**

**if (string.find(sub\_str) == -1):**

**print("NO")**

**else:**

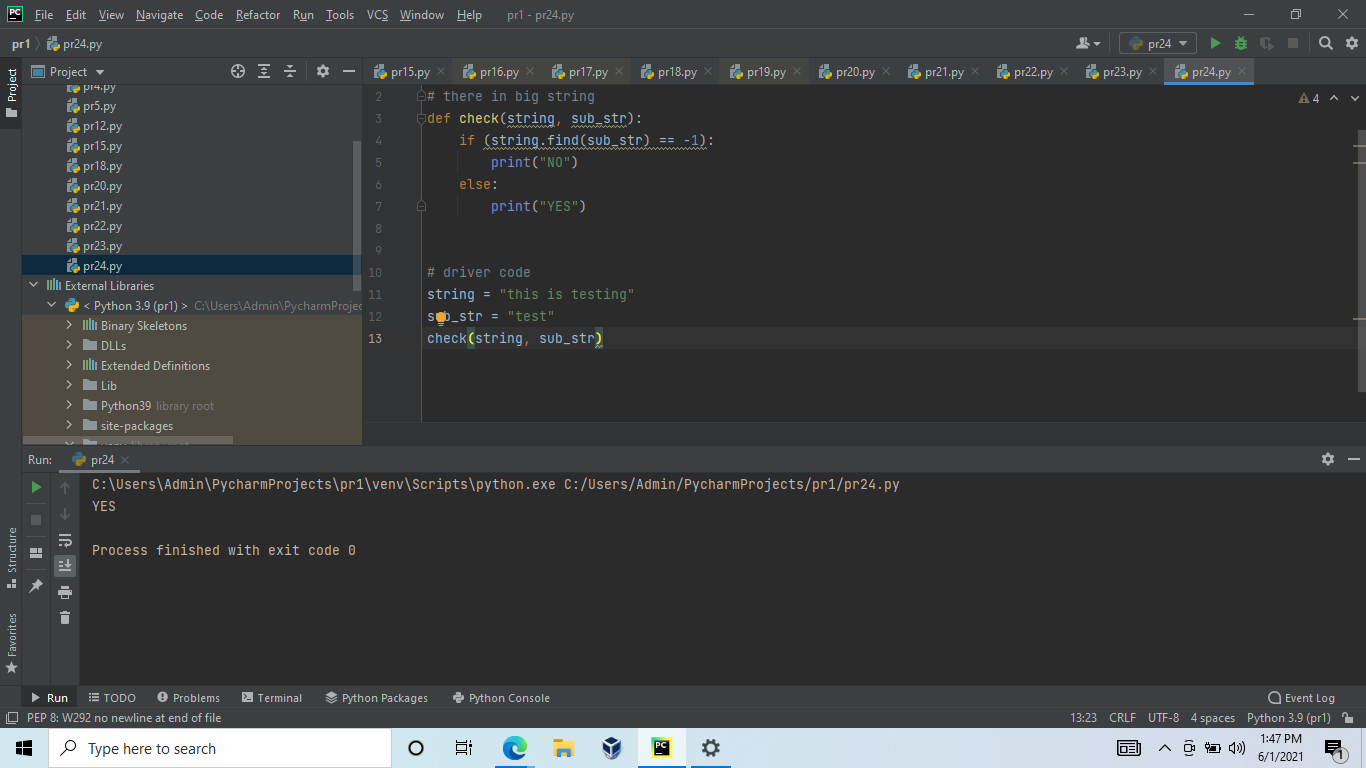
**print("YES")**

**# driver code**

**string = "this is testing"**

**sub\_str = "test"**

**check(string, sub\_str)**

****

**9.Python program to print even length words in a string**

**Given a string. The task is to print all words with even length in the given string.**

**#  even length words in a string**

**def printWords(s):**

**# split the string**

**s = s.split(' ')**

**# iterate in words of string**

**for word in s:**

**# if length is even**

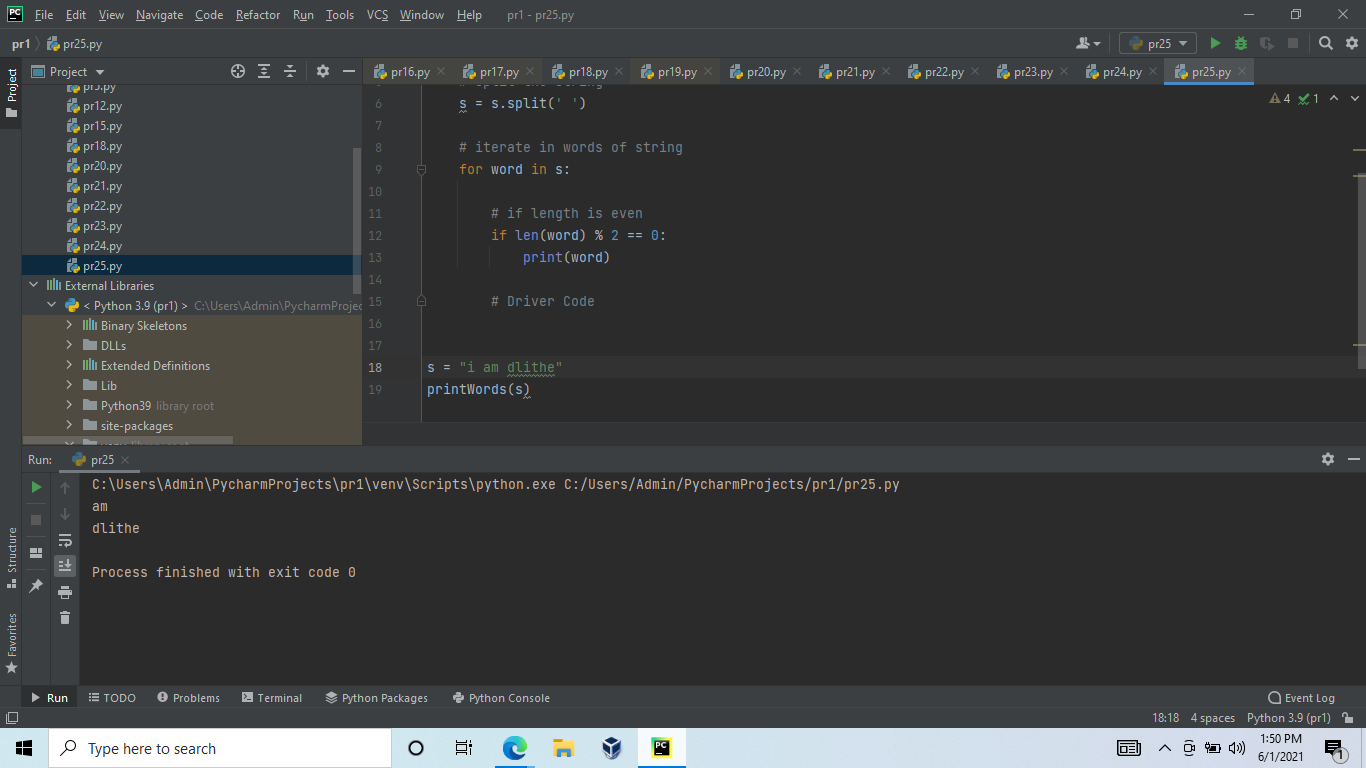
**if len(word) % 2 == 0:**

**print(word)**

**# Driver Code**

**s = "i am dlithe"**

**printWords(s)**

****

**10.Program to check if a string contains any special character**

**Given a string, the task is to check if that string contains any special character (defined special character set). If any special character found, don’t accept that string.**

**# Function checks if the string**

**# contains any special character**

**def run(string):**

**# Make own character set and pass**

**# this as argument in compile method**

**regex = re.compile('[@\_!#$%^&\*()<>?/\|}{~:]')**

**# Pass the string in search**

**# method of regex object.**

**if (regex.search(string) == None):**

**print("String is accepted")**

**else:**

**print("String is not accepted.")**

**# Driver Code**

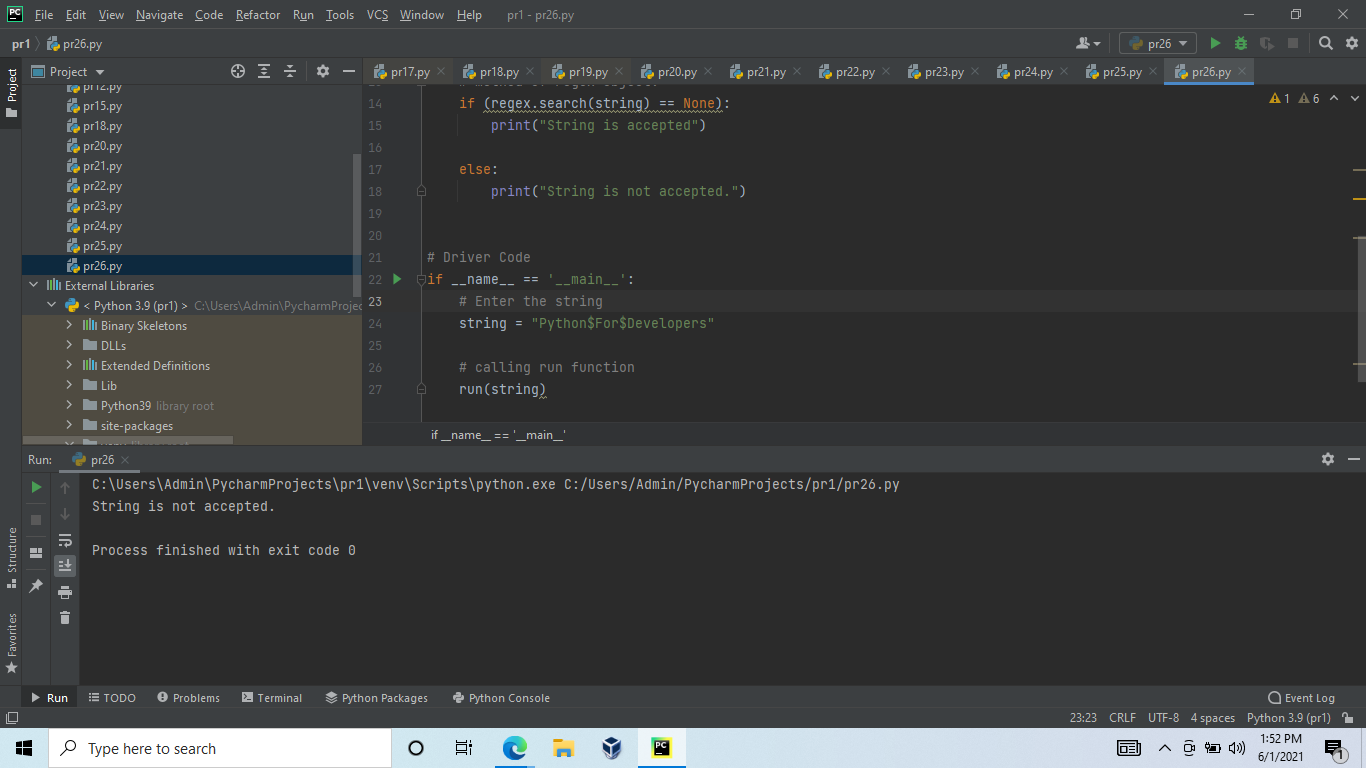
**if \_\_name\_\_ == '\_\_main\_\_':**

**# Enter the string**

**string = "Python$For$Developers"**

**# calling run function**

**run(string)**

****

 Some prime numbers can be expressed as a sum of other consecutive prime numbers.  
  
  
  
  
For example  
  
5 = 2 + 3,  
  
17 = 2 + 3 + 5 + 7,  
  
41 = 2 + 3 + 5 + 7 + 11 + 13.  
  
Your task is to find out how many prime numbers which satisfy this property are present in the range 3 to N subject to a constraint that summation should always start with number 2.  
  
Write code to find out the number of prime numbers that satisfy the above-mentioned property in a given range.  
  
  
  
  
Input Format: First line contains a number N  
  
  
  
  
Output Format: Print the total number of all such prime numbers which are less than or equal to N.  
  
  
  
  
Constraints: 2<N<=12,000,000,000

num = int(input())

arr = []

sum = 0

count = 0

if num > 1:

for i in range(2, num + 2):

for j in range(2, i):

if i % j == 0:

break

else:

arr.append(i)

def is\_prime(sum):

for i in range(2, (sum // 2) +2):

if sum % i == 0:

return False

else:

return True

for i in range(0, len(arr)):

sum = sum + arr[i]

if sum <= num:

if is\_prime(sum):

count = count + 1

print(count)

