



PYTHON ASSIGNMENT

NAME-

PRAJJWAL GUPTA

Answer 1.

```
Nth_fibonacci_no.py X
Nth_fibonacci_no.py >...
1  n = int(input('Enter : '))
2  fibo_nums = [0,1]
3  i=1
4  if(n==1 or n==2):
5      print(n,'th Prime Number is :',fibo_nums[n-1])
6      print('Fibonacci Series :', fibo_nums)
7  elif(n>2):
8      while (True):
9          fib = fibo_nums[i-1]+fibo_nums[i]
10         fibo_nums.append(fib)
11         if(len(fibo_nums)==n):
12             break
13         else:
14             i+=1
15     print(n,'th Fibonacci Number is :', fibo_nums[n-1])
16     print('Fibonacci Series is :', fibo_nums)
17 else:
18     print('Please Enter A Valid Number')
```

PROBLEMS DEBUG CONSOLE **TERMINAL**

```
PS C:\python> python -u "c:\python\Nth_fibonacci_no.py"
Enter : 4
4 th Fibonacci Number is : 2
Fibonacci Series is : [0, 1, 1, 2]
PS C:\python>
```

Answer 2.

```
fibonacci_number.py X
fibonacci_number.py >...
1  n=int(input("Enter the number: "))
2  c=0
3  a=1
4  b=1
5  if n==0 or n==1:
6      print("Yes")
7  else:
8      while c<n:
9          c=a+b
10         b=a
11         a=c
12         if c==n:
13             print("Yes")
14         else:
15             print("No")
16
```

PROBLEMS DEBUG CONSOLE **TERMINAL**

```
PS C:\python> python -u "c:\python\fibonacci_number.py"
Enter the number: 4
No
PS C:\python>
```

Answer 3.

```
nth_multiple_fibonacci.py X
nth_multiple_fibonacci.py > ...
3     f2 = 1
4     i = 2;
5     while i!=0:
6         f3 = f1 + f2;
7         f1 = f2;
8         f2 = f3;
9
10        if f2%k == 0:
11            return n*i
12
13        i+=1
14
15    return
16
17    n = 5;
18    k = 4;
19
20    print("Position of n'th multiple of k in"
21          "Fibonacci Seires is", findPosition(k,n));
22
23
```

PROBLEMS DEBUG CONSOLE **TERMINAL**

```
PS C:\python> python -u "c:\python\nth_multiple_fibonacci.py"
Position of n'th multiple of k inFibonacci Seires is 30
PS C:\python>
```

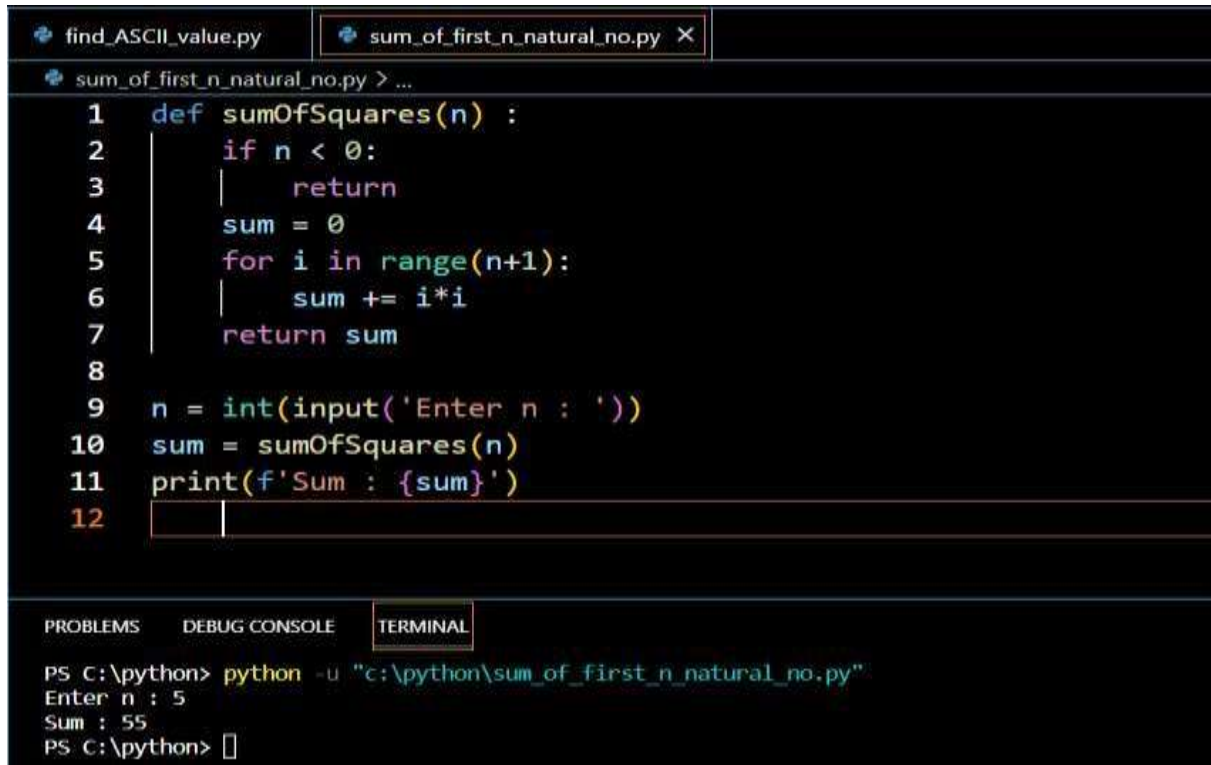
Answer 4.

```
find_ASCII_value.py X
find_ASCII_value.py > ...
1  char="b"
2  print("the ASCII value off", char ,"is",ord(char))
```

PROBLEMS DEBUG CONSOLE **TERMINAL**

```
PS C:\python> python -u "c:\python\find_ASCII_value.py"
the ASCII value off b is 98
PS C:\python>
```

Answer 5.



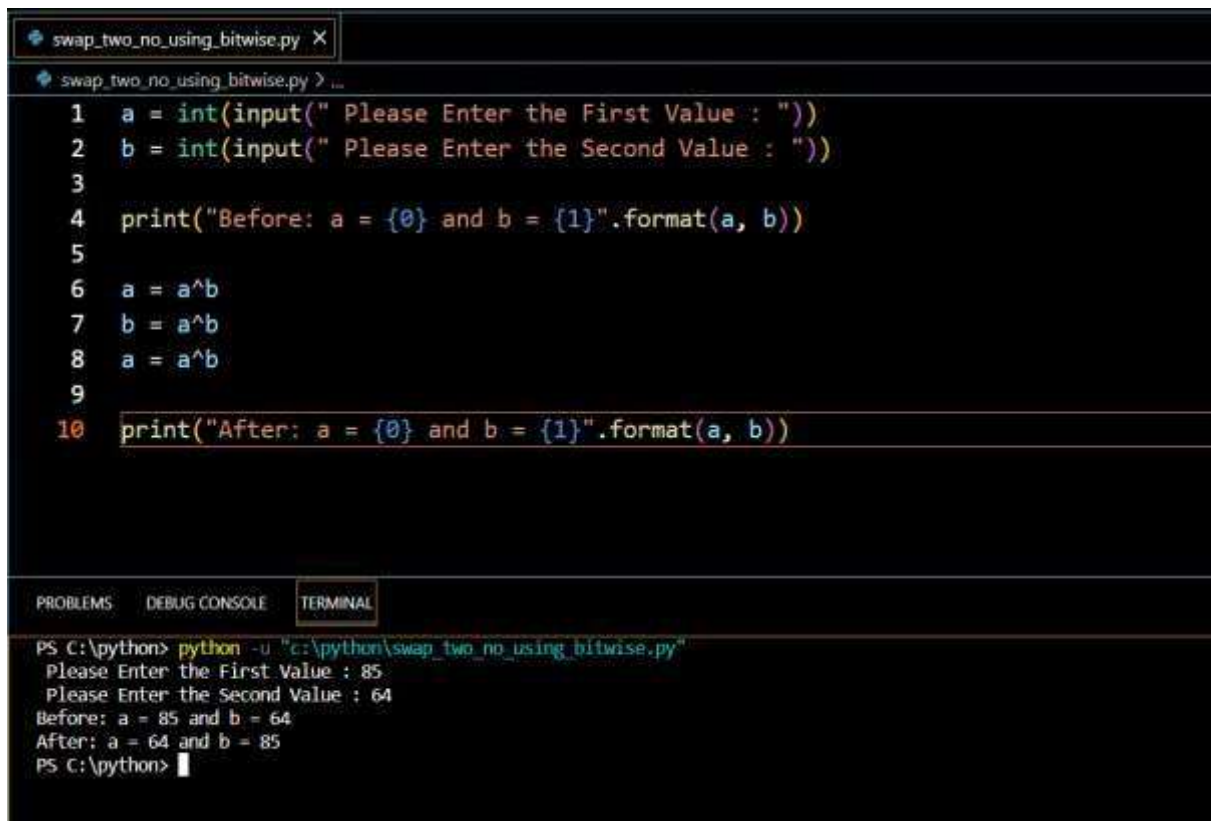
The screenshot shows a code editor with two tabs: `find_ASCII_value.py` and `sum_of_first_n_natural_no.py`. The active tab is `sum_of_first_n_natural_no.py`, which contains the following Python code:

```
1 def sumOfSquares(n) :  
2     if n < 0:  
3         return  
4     sum = 0  
5     for i in range(n+1):  
6         sum += i*i  
7     return sum  
8  
9 n = int(input('Enter n : '))  
10 sum = sumOfSquares(n)  
11 print(f'Sum : {sum}')
```

Below the code editor, the **TERMINAL** tab is active, showing the execution of the script:

```
PS C:\python> python -u "c:\python\sum_of_first_n_natural_no.py"  
Enter n : 5  
Sum : 55  
PS C:\python>
```

Answer 6.



The screenshot shows a code editor with a single tab: `swap_two_no_using_bitwise.py`. The active tab contains the following Python code:

```
1 a = int(input(" Please Enter the First Value : "))  
2 b = int(input(" Please Enter the Second Value : "))  
3  
4 print("Before: a = {0} and b = {1}".format(a, b))  
5  
6 a = a^b  
7 b = a^b  
8 a = a^b  
9  
10 print("After: a = {0} and b = {1}".format(a, b))
```

Below the code editor, the **TERMINAL** tab is active, showing the execution of the script:

```
PS C:\python> python -u "c:\python\swap_two_no_using_bitwise.py"  
Please Enter the First Value : 85  
Please Enter the Second Value : 64  
Before: a = 85 and b = 64  
After: a = 64 and b = 85  
PS C:\python>
```

Answer 7.

```
check_character_is_alphabet.py X check_vowel_or_consonant.py
check_character_is_alphabet.py > _
1 ch = input("Please Enter Your Own Character : ")
2
3 if((ord(ch) >= 65 and ord(ch) <= 90) or (ord(ch) >= 97 and ord(ch) <= 122)):
4     print("The Given Character ", ch, "is an Alphabet")
5 else:
6     print("The Given Character ", ch, "is Not an Alphabet")

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\check_character_is_alphabet.py"
Please Enter Your Own Character : A
The Given Character A is an Alphabet
PS C:\python> python -u "c:\python\check_character_is_alphabet.py"
Please Enter Your Own Character : 5
The Given Character 5 is Not an Alphabet
PS C:\python> 
```

Answer 8.

```
check_character_is_alphabet.py check_vowel_or_consonant.py X
check_vowel_or_consonant.py > _
1 ch = input("Please Enter Your Own Character : ")
2
3 if(ord(ch) == 65 or ord(ch) == 69 or ord(ch) == 73
4     or ord(ch) == 79 or ord(ch) == 85
5     or ord(ch) == 97 or ord(ch) == 101 or ord(ch) == 105
6     or ord(ch) == 111 or ord(ch) == 117):
7     print("The Given Character ", ch, "is a Vowel")
8 elif((ord(ch) >= 97 and ord(ch) <= 122) or (ord(ch) >= 65 and ord(ch) <= 90)):
9     print("The Given Character ", ch, "is a Consonant")

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\check_vowel_or_consonant.py"
Please Enter Your Own Character : a
The Given Character a is a Vowel
PS C:\python> python -u "c:\python\check_vowel_or_consonant.py"
Please Enter Your Own Character : b
The Given Character b is a Consonant
PS C:\python> python -u "c:\python\check_vowel_or_consonant.py"
Please Enter Your Own Character : A
The Given Character A is a Vowel
PS C:\python> python -u "c:\python\check_vowel_or_consonant.py"
Please Enter Your Own Character : B
The Given Character B is a Consonant
PS C:\python> 
```


Answer 9.

```
check_character_is_alphabet.py | check_alphabet_digit_SpecialChar.py X | check_vowel_or_consonant.py
check_alphabet_digit_SpecialChar.py > ...
1 ch = input("Please Enter single character : ")
2
3 if(ord(ch) >= 48 and ord(ch) <= 57):
4 |     print("The Given Character ", ch, "is a Digit")
5 elif((ord(ch) >= 65 and ord(ch) <= 90) or (ord(ch) >= 97 and ord(ch) <= 122)):
6 |     print("The Given Character ", ch, "is an Alphabet")
7 else:
8 |     print("The Given Character ", ch, "is a special character")

PROBLEMS | DEBUG CONSOLE | TERMINAL
PS C:\python> python -u "c:\python\check_alphabet_digit_SpecialChar.py"
Please Enter single character : d
The Given Character  d is an Alphabet
PS C:\python> python -u "c:\python\check_alphabet_digit_SpecialChar.py"
Please Enter single character : 5
The Given Character  5 is a Digit
PS C:\python> python -u "c:\python\check_alphabet_digit_SpecialChar.py"
Please Enter single character : %
The Given Character  % is a special character
PS C:\python> |
```

Answer 10.

```
calculate_percentage_grade.py X | leap_year.py | fibonacci_number.py
calculate_percentage_grade.py > ...
1 Physics = float(input("Please enter Physics Marks: "))
2 Chemistry = float(input("Please enter Chemistry score: "))
3 Biology = float(input("Please enter Biology Marks: "))
4 Mathematics = float(input("Please enter Mathematics Marks: "))
5 Computer = float(input("Please enter Computer Marks: "))
6 total = Physics + Chemistry + Biology + Mathematics + Computer
7 percentage = (total / 500) * 100
8 print("\nTotal Marks = %.2f" %total)
9 print("Marks Percentage = %.2f" %percentage)
10 if(percentage>=90):
11 |     print("Grade: A")
12 elif(percentage>=80 and percentage<90):
13 |     print("Grade: B")
14 elif(percentage>=70 and percentage<80):
15 |     print("Grade: C")
16 elif(percentage>=60 and percentage<70):
17 |     print("Grade: D")
18 elif(percentage>=60 and percentage<40):
19 |     print("grade: E")
20 else:
21 |     print("Grade: F")

PROBLEMS | DEBUG CONSOLE | TERMINAL
Please enter Physics Marks: 50
Please enter Chemistry score: 49
Please enter Biology Marks: 67
Please enter Mathematics Marks: 81
Please enter Computer Marks: 97
Marks Percentage = 68.80
Grade: D
```

Answer 11.

```
check_strong_no_or_not.py calculate_gross_salary.py X
calculate_gross_salary.py > ...
1 sal=float(input('Enter the basic salary:'))
2 if(sal<=10000):
3     hra=0.2*sal
4     da=0.8*sal
5     Salary=sal+hra+da
6 elif(sal<=20000):
7     hra=0.25*sal
8     da=0.9*sal
9     Salary=hra+da+sal
10 else:
11     hra=0.3*sal
12     da=0.95*sal
13     Salary=hra+da+sal
14 print('The gross salary is',Salary)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\calculate_gross_salary.py"
Enter the basic salary:4000
The gross salary is 8000.0
PS C:\python> 
```

Answer 12.

```
check_strong_no_or_not.py calculate_electric_bill.py X
calculate_electric_bill.py > ...
1 x=float(input('Enter the electricity unit'))
2 if(x<=50):
3     bill=0.5*x
4 if(x>50 and x<=150):
5     bill=50*0.5+(x-50)*0.75
6 if(x>150 and x<=250):
7     bill=50*0.5+100*0.75+(x-150)*1.20
8 if(x>250):
9     bill=50*0.5+100*0.75+100*1.2+(x-250)*1.5
10 Total=bill+0.2*bill
11 print('Total bill after surcharge is',Total)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\calculate_electric_bill.py"
Enter the electricity unit56
Total bill after surcharge is 35.4
PS C:\python> 
```

Answer 13.

```
calculate_percentage_grade.py  a_to_z_using_while_loop.py x  leap_year.py
a_to_z_using_while_loop.py > ...
1  def printalphabet():
2      i = 0
3      while i < 26:
4          print(chr(97 + i), end = " ")
5          i = i + 1
6
7  printalphabet()
8

PROBLEMS  DEBUG CONSOLE  TERMINAL

PS C:\python> python -u "c:\python\a_to_z_using_while_loop.py"
a b c d e f g h i j k l m n o p q r s t u v w x y z
PS C:\python>
```

Answer 14.

```
calculate_percentage_grade.py  a_to_z_using_while_loop.py  Ist_and_last_digit_of_no.py x
Ist_and_last_digit_of_no.py > ...
1  def firstDigit(n) :
2      while n >= 10:
3          n = n / 10;
4      return int(n)
5  def lastDigit(n) :
6      return (n % 10)
7  n =int(input("enter the number: "));
8  print(firstDigit(n), end = " ")
9  print(lastDigit(n))
10

PROBLEMS  DEBUG CONSOLE  TERMINAL

PS C:\python> python -u "c:\python\Ist_and_last_digit_of_no.py"
enter the number: 468731
4 1
PS C:\python> |
```

Answer 15.


```
sum_of_digit_of_no.py X ATM_machine_logic.py
sum_of_digit_of_no.py > ...
1 num = int(input("Enter a number: "))
2 sum = 0
3 while num > 0:
4     d = num%10
5     num = num//10
6     sum += d
7 print("The sum of digits of number is",sum)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\sum_of_digit_of_no.py"
Enter a number: 123456987
The sum of digits of number is 45
PS C:\python> 
```

Answer 16.

```
product_of_digit_of_no.py X
product_of_digit_of_no.py > ...
1 num = int(input("Enter any number : "))
2 temp = num
3 product = 1;
4 while(temp != 0):
5     product = product * (temp % 10);
6     temp = int(temp / 10)
7 print("Product of all digits in", num, ":", product)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\product_of_digit_of_no.py"
Enter any number : 5613
Product of all digits in 5613 : 90
```

Answer 17.

```
product_of_digit_of_no.py  print_reverse_of_number.py X
print_reverse_of_number.py > ...
1  n=int(input("Enter number: "))
2  rev=0
3  while(n>0):
4      dig=n%10
5      rev=rev*10+dig
6      n=n//10
7  print("Reverse of the number:",rev)

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\print_reverse_of_number.py"
Enter number: 324689354
Reverse of the number: 453986423
PS C:\python> 
```

Answer 18.

```
pallindrome.py.py X
pallindrome.py.py > ...
17  n=int(input("enter the number"))
18  if str(n)==str(n)[::-1]:
19      print("pallindrome")
20  else:
21      print("not pallindrome")

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\pallindrome.py.py"
enter the number12321
pallindrome
PS C:\python> 
```

Answer 19.

```
factor_of_a_number.py X
factor_of_a_number.py > print_factors
1 def print_factors(n):
2     for i in range(1, n+1):
3         if n % i == 0:
4             print(i)
5 number = int(input("Enter a number : "))
6
7 print("The factors for {} are : ".format(number))
8 print_factors(number)

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\factor_of_a_number.py"
Enter a number : 25
The factors for 25 are :
1
5
25
PS C:\python> 
```

Answer 20.

```
prime_number.py  factorial_of_number.py X
factorial_of_number.py > ...
1 num = int(input("Enter a number: "))
2 factorial = 1
3
4 if num < 0:
5     print("Sorry, factorial does not exist for negative numbers")
6 elif num == 0:
7     print("The factorial of 0 is 1")
8 else:
9     for i in range(1,num + 1):
10         factorial = factorial*i
11     print("The factorial of",num,"is",factorial)

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\factorial_of_number.py"
Enter a number: 5
The factorial of 5 is 120
PS C:\python> 
```

Answer 21.

```
HCF.PY X
HCF.PY > ...
1 x = int(input('Enter First Number: '))
2 y = int(input('Enter Second Number: '))
3 if x > y:
4     smaller = y
5 else:
6     smaller = x
7 for i in range(1, smaller+1):
8     if((x % i == 0) and (y % i == 0)):
9         hcf = i
10 print('The hcf of', x, 'and', y, 'is', hcf)

PROBLEMS  DEBUG CONSOLE  TERMINAL

PS C:\python> python -u "c:\python\HCF.PY"
Enter First Number: 56
Enter Second Number: 24
The hcf of 56 and 24 is 8
PS C:\python> 
```

Answer 22.

```
LCM.PY X
LCM.PY > ...
1 num1 = int(input("enter the first number: "))
2 num2 = int(input("enter the second number: "))
3 for i in range(max(num1, num2), 1 + (num1 * num2)):
4     if i % num1 == i % num2 == 0:
5         lcm = i
6         break
7 print("LCM of", num1, "and", num2, "is", lcm)

PROBLEMS  DEBUG CONSOLE  TERMINAL

PS C:\python> python -u "c:\python\LCM.PY"
enter the first number: 56
enter the second number: 24
LCM of 56 and 24 is 168
PS C:\python> 
```

Answer 23.


```
LCM.PY prime_number.py X
prime_number.py > ...
1 n=int(input("enter the number= "))
2 for i in range(2,n):
3     if n%i==0:
4         print("number is not a prime")
5         break
6 else:
7     print("number is a prime number")

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\prime_number.py"
enter the number= 27
number is not a prime
PS C:\python> 
```

Answer 24.

```
print_prime_no_from_1_to_n.py X
print_prime_no_from_1_to_n.py > ...
1 Number = int(input(" Please Enter last digit till you want to find prime number: "))
2 print("Prime numbers between", 1, "and", Number, "are:")
3 for num in range(1, Number + 1):
4     if num > 1:
5         for i in range(2, num):
6             if (num % i) == 0:
7                 break
8         else:
9             print(num)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\print_prime_no_from_1_to_n.py"
Please Enter last digit till you want to find prime number: 8
Prime numbers between 1 and 8 are:
2
3
5
7
PS C:\python> 
```

Answer 25.

```
sum_of_prime_1_to_n.py X
sum_of_prime_1_to_n.py > ...
1 x=int(input("Please enter the maximum value: "))
2 total=0
3 for Number in range(1,x+1):
4     count=0;
5     for i in range(2,(Number//2+1)):
6         if(Number%i==0):
7             count=count+1
8             break
9     if(count==0 and Number !=1):
10        total=total+Number
11 print("Sum of prime numbers from 1 to %d=%d"%(x,total))

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\sum_of_prime_1_to_n.py"
Please enter the maximum value: 21
Sum of prime numbers from 1 to 21=77
PS C:\python> █
```

Answer 26.

```
prime_factor_of_number.py X
prime_factor_of_number.py > ...
1 n = int(input("Enter the number for calculating the prime factors :"))
2 for i in range(2,n + 1):
3     if n % i == 0:
4         count = 1
5         for j in range(2,(i//2 + 1)):
6             if(i % j == 0):
7                 count = 0
8                 break
9         if(count == 1):
10            print(i)

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\prime_factor_of_number.py"
Enter the number for calculating the prime factors :84
2
3
7
PS C:\python> █
```

Answer 27.

```
Armstrong_no_or_not.py X Armstrong_no_1_to_n.py
Armstrong_no_or_not.py > ...
1 x=int(input("Enter any Number"))
2 sum=0
3 tmp=x
4 while x>0:
5     digit=(x%10)
6     sum=sum+digit**3
7     x=x//10
8 if sum==tmp:
9     print(tmp, " is Armstrong Number")
10 else:
11     print(tmp, " is not Armstrong Number")

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\Armstrong_no_or_not.py"
Enter any Number153
153 is Armstrong Number
PS C:\python> █
```

Answer 28.

```
Armstrong_no_or_not.py Armstrong_no_1_to_n.py X
Armstrong_no_1_to_n.py > ...
1 lower = int(input("Enter lower range: "))
2 upper = int(input("Enter upper range: "))
3 print("The armstrong numbers are: ")
4 for number in range(lower, upper + 1):
5     order = len(str(number))
6     sum_pow = 0
7
8     temp = number
9     while temp:
10         temp,digit = divmod(temp,10)
11         sum_pow+=digit ** order
12     if number == sum_pow:
13         print(number)
14

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\Armstrong_no_1_to_n.py"
Enter lower range: 100
Enter upper range: 1000
The armstrong numbers are:
153
370
371
407
PS C:\python> █
```

Answer 29.

```
check_perfect_no_or_not.py X
check_perfect_no_or_not.py > ...
1 Number = int(input(" Please Enter any Number: "))
2 Sum = 0
3 for i in range(1, Number):
4     if(Number % i == 0):
5         Sum = Sum + i
6 if (Sum == Number):
7     print(" %d is a Perfect Number" %Number)
8 else:
9     print(" %d is not a Perfect Number" %Number)

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\check_perfect_no_or_not.py"
Please Enter any Number: 28
28 is a Perfect Number
PS C:\python> 
```

Answer 30.

```
check_strong_no_or_not.py X
check_strong_no_or_not.py > sum
1 sum=0
2 num=int(input("Enter a number:"))
3 temp=num
4 while(num):
5     i=1
6     fact=1
7     rem=num%10
8     while(i<=rem):
9         fact=fact*i
10        i=i+1
11    sum=sum+fact
12    num=num//10
13 if(sum==temp):
14     print("Given number is a strong number")
15 else:
16     print("Given number is not a strong number")

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\check_strong_no_or_not.py"
Enter a number:145
Given number is a strong number
PS C:\python> 
```

Answer 31.


```
check_string_is_symetrical_or_pallindrome.py X reverse_word_in_string.py remove_i'th_character_from_string.py
check_string_is_symetrical_or_pallindrome.py > ...
3 def is_symmetrical(s):
4     n = len(s)
5     for i in range(n // 2):
6         if s[i] != s[n - i - 1]:
7             return False
8     return True
9
10 def is_palindrome(s):
11     return s == s[::-1]
12
13 string = input("Enter a string: ")
14
15 if is_symmetrical(string):
16     print("The string is symmetrical")
17 else:
18     print("The string is not symmetrical")
19
20 if is_palindrome(string):
21     print("The string is a palindrome")
22 else:
23     print("The string is not a palindrome")

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\check_string_is_symetrical_or_pallindrome.py"
Enter a string: ronaldo
The string is not symmetrical
The string is not a palindrome
PS C:\python> 
```

Answer 32.

```
check_string_is_symetrical_or_pallindrome.py reverse_word_in_string.py X remove_i'th_character_from_string.py
reverse_word_in_string.py > reverse_words
3 def reverse_words(string):
4     words = string.split()
5     reversed_words = words[::-1]
6     reversed_string = " ".join(reversed_words)
7     return reversed_string
8
9 string = input("Enter a string: ")
10 reversed_string = reverse_words(string)
11 print("Reversed string: ", reversed_string)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\reverse_word_in_string.py"
Enter a string: cristiano ronaldo
Reversed string:  ronaldo cristiano
PS C:\python> 
```

Answer 33.

count_no_of_matching_char_in_a_pair_of_string.py remove_i'th_character_from_string.py X

remove_i'th_character_from_string.py > ...

```
1 # Ways to remove i'th character from string in Python
2 #     using slicing
3 def remove_char(string, i):
4     |     return string[:i] + string[i+1:]
5
6 string = "Hello World"
7 new_string = remove_char(string, 4)
8 print(new_string)
9 #     using string replace
10 def remove_char(string, i):
11     |     return string.replace(string[i], "", 1)
12
13 string = "Hello World"
14 new_string = remove_char(string, 4)
15 print(new_string)
16 #     using a loop
17 def remove_char(string, i):
18     |     new_string = ""
19     |     for j in range(len(string)):
20     |         |     if j != i:
21     |         |         new_string += string[j]
22     |     return new_string
23
24 string = "Hello World"
25 new_string = remove_char(string, 4)
26 print(new_string)
27
```

PROBLEMS DEBUG CONSOLE TERMINAL

```
PS C:\python> python -u "c:\python\remove_i'th_character_from_string.py"
Hell World
Hell World
Hell World
PS C:\python>
```

Answer 34.

```
check_substring_present_in_string.py X count_word_frequency_in_string.py convert_snake_case_to_pascal_c
check_substring_present_in_string.py > ...
3 def is_substring(string, substring):
4     if substring in string:
5         return True
6     else:
7         return False
8
9 string = input("Enter a string: ")
10 substring = input("Enter a substring: ")
11
12 if is_substring(string, substring):
13     print("The substring is present in the string")
14 else:
15     print("The substring is not present in the string")

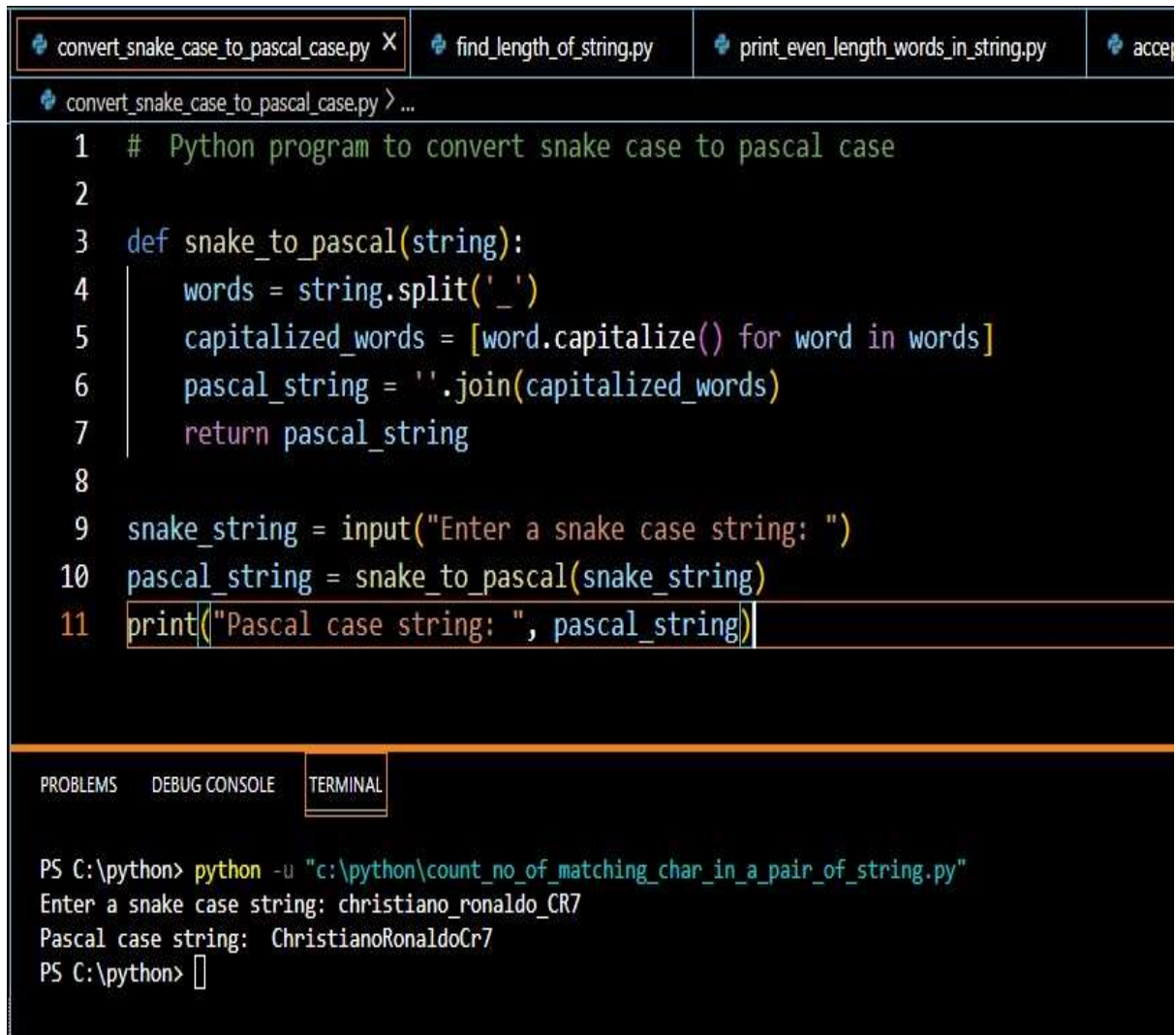
PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\check_substring_present_in_string.py"
Enter a string: christiano ronaldo
Enter a substring: ronaldo
The substring is present in the string
PS C:\python> 
```

Answer 35.

```
count_word_frequency_in_string.py X convert_snake_case_to_pascal_case.py find_length_of_string.py print_even
count_word_frequency_in_string.py > ...
1 # Python program to count words frequency in String Shorthands
2
3 from collections import Counter
4 def count_words(string):
5     words = string.split()
6     word_counts = Counter(words)
7     return word_counts
8
9 string = input("Enter a string: ")
10 word_counts = count_words(string)
11
12 print("Word frequencies:")
13 for word, count in word_counts.items():
14     print(word, ":", count)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\count_no_of_matching_char_in_a_pair_of_string.py"
Enter a string: pake ped pe pake papita pake ped ya pake papita
Word frequencies:
pake : 4
ped : 2
pe : 1
papita : 2
ya : 1
PS C:\python> 
```

Answer 36.



The image shows a code editor with four tabs: `convert_snake_case_to_pascal_case.py` (active), `find_length_of_string.py`, `print_even_length_words_in_string.py`, and `accept...`. The active tab contains the following Python code:

```
1 # Python program to convert snake case to pascal case
2
3 def snake_to_pascal(string):
4     words = string.split('_')
5     capitalized_words = [word.capitalize() for word in words]
6     pascal_string = ''.join(capitalized_words)
7     return pascal_string
8
9 snake_string = input("Enter a snake case string: ")
10 pascal_string = snake_to_pascal(snake_string)
11 print("Pascal case string: ", pascal_string)
```

Below the code editor is a terminal window with the following output:

```
PS C:\python> python -u "c:\python\count_no_of_matching_char_in_a_pair_of_string.py"
Enter a snake case string: cristiano_ronaldo_CR7
Pascal case string: CristianoRonaldoCr7
PS C:\python> 
```


Answer 37.

find_length_of_string.py X practice.py 3 print_even_length_words_in_string.py accept_s

find_length_of_string.py > ...

```
2 #           using len() function
3 string = input("Enter a string: ")
4 length = len(string)
5 print("Length of the string:", length)
6 #           using loop
7 string = input("Enter a string: ")
8 length = 0
9 for char in string:
10 |     length += 1
11 print("Length of the string:", length)
12 #           Using the sys.getsizeof() function
13 import sys
14 string = input("Enter a string: ")
15 length = sys.getsizeof(string) - sys.getsizeof('')
16 print("Length of the string:", length)
17 #           Using the count() method
18 string = "Hello, World!"
19 length = string.count('')
20 print(length - 1)
21
```

PROBLEMS (3) DEBUG CONSOLE TERMINAL

```
PS C:\python> python -u "c:\python\find_length_of_string.py"
Enter a string: ronaldo
Length of the string: 7
Enter a string: 
```

Answer 38.

```
print_even_length_words_in_string.py X accept_string_which_contain_all_vowel.py count_no_of_matching_char_i
print_even_length_words_in_string.py > ...
1 # Python program to print even length words in a string
2
3 def print_even_length_words(string):
4     words = string.split()
5     for word in words:
6         if len(word) % 2 == 0:
7             print(word)
8
9 string = str(input("enter the string: "))
10 print_even_length_words(string)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\print_even_length_words_in_string.py"
enter the string: he is going to marcket
he
is
to
PS C:\python> |
```

Answer 39.

```
accept_string_which_contain_all_vowel.py X count_no_of_matching_char_in_a_pair_of_string.py
accept_string_which_contain_all_vowel.py > _
1 # Python program to accept the strings which contains all vowels
2
3 def contains_all_vowels(string):
4     string = string.lower()
5
6     if 'a' in string and 'e' in string and 'i' in string and 'o' in string and 'u' in string:
7         return True
8     else:
9         return False
10 string1 = str(input("enter the string 1: "))
11 string2 = str(input("enter the string 2: "))
12 string3 = str(input("enter the string 3: "))
13 print(contains_all_vowels(string1))
14 print(contains_all_vowels(string2))
15 print(contains_all_vowels(string3))

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\accept_string_which_contain_all_vowel.py"
enter the string 1: aeiou
enter the string 2: aeibc
enter the string 3: xyz
True
False
False
PS C:\python> |
```

Answer 40.

```
count_no_of_matching_char_in_a_pair_of_string.py X
count_no_of_matching_char_in_a_pair_of_string.py > ...

1 # Python program to count the Number of matching characters in a pair of string
2
3 def count_matching_chars(str1, str2):
4     set1 = set(str1)
5     set2 = set(str2)
6     matching_chars = set1.intersection(set2)
7     return len(matching_chars)
8
9 str1 = str(input("enter the string 1: "))
10 str2 = str(input("enter the string 2: "))
11 print(count_matching_chars(str1, str2))

PROBLEMS  DEBUG CONSOLE  TERMINAL

PS C:\python> python -u "c:\python\count_no_of_matching_char_in_a_pair_of_string.py"
enter the string 1: he is a good man
enter the string 2: his name is ran
8
PS C:\python> |
```

Answer 41.

```
check_strong_no_or_not.py  remove_duplicate_from_string.py X
remove_duplicate_from_string.py > ...

3 strValue =str(input("enter the string you want: "))
4 strValue = ''.join(sorted(set(strValue), key=strValue.index))
5 print(strValue)

PROBLEMS  DEBUG CONSOLE  TERMINAL

PS C:\python> python -u "c:\python\remove_duplicate_from_string.py"
enter the string you want: christuano ronaldo
christuano ld
PS C:\python> |
```

Answer 42.

```
count_least_frequencychar.py X remove_duplicate_from_string.py
count_least_frequencychar.py > ...
1 a=str(input("enter the string: "))
2 dict ={}
3
4 for character in a:
5     if character in dict:
6         dict[character]+=1
7     else:
8         dict[character]=1
9
10 print("The least frequent character is", str(min(dict, key = dict.get)))

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\count_least_frequencychar.py"
enter the string: bbsvssvr
The least frequent character is r
PS C:\python> 
```

Answer 43.

```
count_maximum_frequency_in_char.py X
count_maximum_frequency_in_char.py > ...
1 # Python programs to count maximum frequency character in String
2
3 a=str(input("enter the string: "))
4 dict ={}
5
6 for character in a:
7     if character in dict:
8         dict[character]-=1
9     else:
10        dict[character]=1
11
12 print("The least frequent character is", str(min(dict, key = dict.get)))

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\count_maximum_frequency_in_char.py"
enter the string: ghagsiuofhg
The least frequent character is g
PS C:\python> 
```

Answer 44.


```
check_string_contain_any_special_char.py X
check_string_contain_any_special_char.py > ...

3 import re
4 string = input('Enter any string: ')
5 special_char = re.compile('[@_!#$%^&*()<>?/\|}{~:~:]')
6 if(special_char.search(string) == None):
7     print('String does not contain any special characters.')
8 else:
9     print('The string contains special characters.')

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\check_string_contain_any_special_char.py"
Enter any string: ^bkcxl
The string contains special characters.
PS C:\python> 
```

Answer 45.

```
split_and_join_string.py X
split_and_join_string.py > ...

1 s = str(input("enter the string: "))
2 print(s.split(" "))
3 print("-".join(s.split()))

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\split_and_join_string.py"
enter the string: split and join string
['split', 'and', 'join', 'string']
split-and-join-string
PS C:\python> 
```

Answer 46.

```
find_uncommon_word_from_2_string.py X
find_uncommon_word_from_2_string.py > ...
1 str1 = input("Enter first string : ")
2 str2 = input("Enter second string : ")
3 def uncomn_wrd(x,y):
4     x = x.split()
5     y = y.split()
6     k = set(x).symmetric_difference(set(y))
7     return k
8 print("Uncommon words are :", list(uncomn_wrd(str1, str2)))

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\find_uncommon_word_from_2_string.py"
Enter first string : find uncommon
Enter second string : find common
Uncommon words are : ['uncommon', 'common']
PS C:\python> 
```

Answer 47.

```
string_slicing_to_rotate_a_string.py  replace_duplicate_occurance_of_string.py X
replace_duplicate_occurance_of_string.py > ...
3 s=input("enter a string: ")
4 i=0
5 s1=""
6 for x in s:
7     if s.index(x)==i:
8         s1+=x
9         i+=1
10 print(s1)

PROBLEMS  DEBUG CONSOLE  TERMINAL
PS C:\python> python -u "c:\python\replace_duplicate_occurance_of_string.py"
enter a string: aabaahabcchs
abhcs
PS C:\python> 
```

Answer 48.

```
string_slicing_to_rotate_a_string.py X replace_duplicate_occurance_of_string.py
string_slicing_to_rotate_a_string.py > ...
1 # String slicing in Python to rotate a string
2
3 input_str = input("Enter a string: ")
4 rotate_by = int(input("Enter the number of positions to rotate: "))
5
6 rotated_str = input_str[rotate_by:] + input_str[:rotate_by]
7
8 print("Original string:", input_str)
9 print("Rotated string:", rotated_str)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\string_slicing_to_rotate_a_string.py"
Enter a string: slicing
Enter the number of positions to rotate: 2
Original string: slicing
Rotated string: icingsl
PS C:\python> |
```

Answer 49.

```
string_slicing_to_rotate_a_string.py find_duplicate_character_in_sting.py X replace_duplicate_occurance_of_string.py
find_duplicate_character_in_sting.py > ...
1 input_str = input("Enter a string: ")
2
3 char_freq = {}
4 for char in input_str:
5     if char in char_freq:
6         char_freq[char] += 1
7     else:
8         char_freq[char] = 1
9
10 print("Duplicate characters in the string:")
11 for char in char_freq:
12     if char_freq[char] > 1:
13         print(char)

PROBLEMS DEBUG CONSOLE TERMINAL
PS C:\python> python -u "c:\python\find_duplicate_character_in_sting.py"
Enter a string: cristiano ronaldo
Duplicate characters in the string:
r
i
a
n
o
PS C:\python> |
```

Answer 50.

replace_occurance_of_substring.py X

replace_occurance_of_substring.py > ...

```
3 input_str = input("Enter a string: ")
4 sub_str = input("Enter the substring to replace: ")
5 replace_str = input("Enter the replacement string: ")
6
7 new_str = input_str.replace(sub_str, replace_str)
8
9 print("Original string:", input_str)
10 print("New string:", new_str)
```

PROBLEMS DEBUG CONSOLE TERMINAL

```
PS C:\python> python -u "c:\python\replace_occurance_of_substring.py"
Enter a string: cristiano ronaldo
Enter the substring to replace: ronaldo
Enter the replacement string: CR7
Original string: cristiano ronaldo
New string: cristiano CR7
PS C:\python> 
```
