

1 Hour Paper 4:00 PM-5:00 PM

Python - Daily Test 11 - 8-Feb-17

1. Write a program in Python to find perfect squares in user entered range.

Given two given numbers num1 and num2 where  $1 \leq \text{num1} \leq \text{num2}$ , find the number of perfect squares between num1 and num2 (num1 and num2 inclusive). Implement as follows:

- \* Accept num1 and num2 from user.
- \* If either of the num1 or num2 is negative or  $\text{num1} > \text{num2}$  don't pass the values to function.
- \* Implement the functionality of perfect squares numbers in function by accepting num1 and num2. Take default values of num1 as 1 and num2 as 100, if values not passed. Return the perfect numbers in list.
- \* In main program, get the list from function and print the output.

Following is sample, please write generic code:

Enter Number 1: 9

Enter Number 2: 25

Perfect squares between 9 and 25 are [9, 16, 25] and total count is 3

Enter Number 1: -20

Enter Number 2: 100

Perfect squares between 1 and 100 are [1, 4, 9, 16, 25, 36, 49, 64, 81, 100] and total count is 10

Enter Number 1: 20

Enter Number 2: -50

Perfect squares between 1 and 100 are [1, 4, 9, 16, 25, 36, 49, 64, 81, 100] and total count is 10

Enter Number 1: 100

Enter Number 2: 20

Perfect squares between 1 and 100 are [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]  
and total count is 10

Enter Number 1: 1

Enter Number 2: 1000

Perfect squares between 1 and 1000 are [1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961] and total count is 31

2. Write a program in Python to find the bit difference. Perform in the following way:

- \* Write a function, say, bit\_difference(), to accept two numbers, say, num1 and num2. The function need to count the number of bits needed to be flipped to convert num1 to num2. Return the number of bits needed.

- \* Accept number, say num1, from user

- \* Accept number, say num2, from user

- \* In the main, call the function with the values, num1 and num2, get the result of number of bits needed to flip and print it. Call the function using keyword arguments.

Following is sample, write only generic code,

Enter Number 1: 10

Enter Number 2: 20

The number of bits needed to flip 10 to 20 is 4.

NOTE: Following is the way to add zeros in front of number, if required.

```
num = 57
```

```
num = '{:08b}'.format(num)
```

```
print type(num), ",", num #Will give --> <type 'str'>, 00111001
```

3. Write a program in Python to perform maximum product of two numbers

as follows (example is only for understanding, need to write generic code. You can take values as below but code should be generic)

- \* Implement function called `maximum_product_two_numbers()`, which accepts two arguments, first one is the length of list and second one is `elems`, which is variable-length argument. In other words, the function need to accept variable length arguments or variable number of arguments i.e., first argument is length of variable arguments, second argument is elements themselves. The function need to return product of first two maximum numbers.

- \* Call from main program such as

Ex:

```
maximum_product_two_numbers(5, 1, 100, 42, 4, 23)
```

or

```
maximum_product_two_numbers(9, 1, 100, 42, 4, 23, 123, 45, 92, 99)
```

- \* Print the received value from function, which is maximum product.

4. Write a program in Python to count maximum 0's between two immediate 1's.

- \* Accept a number, say `num`, from user

- \* Pass this number to function, say `max_zeros_cnt()`.

- \* Return the result from function and print the result.

- \* The task of this function is to find the maximum 0's between two immediate 1's in binary representation of given number. Return -1 if binary representation contains less than two 1's.

Following are few samples, write generic code only:

Input :

Enter Number: 47

Output:

Maximum zeros between two 1's for number 47 is 1

Explanation:

Binary representation of num 47 is 101111 and there is only 1 zero between 1's

Input :

Enter Number: 549

Output:

Maximum zeros between two 1's for number 549 is 3

Explanation:

Binary representation of num 549 is 1000100101 and maximum number of zero's between 1's are 3 (from 3, 2, 1, => 3 is maximum)

Input :

Enter Number: 1030

Output:

Maximum zeros between two 1's for number 1030 is 7

Explanation:

Binary representation of num 1030 is 10000000110 and maximum number of zero's between 1's are 7

Input :

Enter Number: 8

Output:

Maximum zeros between two 1's for number 8 is -1

Explanation:

Binary representation of num 8 is 1000 and there are no zero's between 1's, so to return -1, representing not found.