

## Control Statements

1. Write the program in Python to perform the following (using while loop):
  - To accept user input (integers only) until user enters zero (zero is end of input). Sum all the user entered numbers and display the sum.
  - To accept user input (strings only) until user enters “end” or “END” (signifying the end of input). Append all the user entered strings and display the final string.

**NOTE:** Should not use condition as True (infinite loop), use proper condition only.
2. Write a program in Python to check whether user entered number is Armstrong number or not. User can enter any digits number i.e., either 1 or 2 or 3 or 4 or so on digits. So, write only generic code.

**About Armstrong Number**

An n-digit number that is the sum of the nth powers of its digits is called an Armstrong number. The n-digit numbers equal to the sum of the powers of their digits (a finite sequence) are called Armstrong numbers and are given by 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, ..

An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since  $3^3 + 7^3 + 1^3 = 371$ .

An Armstrong number of four digits is an integer such that the sum of the power of 4 of its digits is equal to the number itself. For example, 1634 is an Armstrong number since  $1^4 + 6^4 + 3^4 + 4^4 = 1634$ .
3. Write a program in Python to find smallest factorial number containing given trailing zeros. Given a number num. The task is to find the smallest number whose factorial contains at least num trailing zeroes.

NOTE: Don't use functions and infinite loops.

Examples:

Input : num = 1  
Output : 5  
1!, 2!, 3!, 4! does not contain trailing zero.  
5! = 120, which contains one trailing zero.

Input : num = 6  
Output : 25  
25! = 15511210043330985984000000
4. Write the program in Python to perform the following:
  - Take three empty lists to store integers, floats and strings.
  - To accept user input (integers or float or strings) continuously until user either enters zero or “end” or “END” (zero, “end” or “END” signifies end of input)
  - As per the input in the previous step add to respective list
  - Once user enters all the inputs, display the lists of integers, floats and strings.

**NOTE:** Should not use condition as True (infinite loop), use proper condition only.

5. Write a program in Python to perform the following:  
Given a number num, find the last non-zero digit in num!

Examples:

Input : num = 5

Output : 2

$5! = 5 * 4 * 3 * 2 * 1 = 120$

Last non-zero digit in 120 is 2.

Input : num = 33

Output : 8

**NOTE:** Don't use functions and infinite loops

6. Write a program in Python to perform the following:
- Accept user input of +ve and -ve values and store it in list, say mylist (End with 0 or number of numbers)
  - Move all negative elements to end.

Given an unsorted list having both negative and positive integers. The task is to place all negative element at the end of list without changing the order of positive element and negative element. No need to modify the existing list, you can create new list for output.

Examples (Code should be generic, this is just an example)

Input : mylist = [1, -1, 3, 2, -7, -5, 11, 6 ]

Output : [1 3 2 11 6 -1 -7 -5]

Input : mylist= [-5, 7, -3, -4, 9, 10, -1, 11]

Output : [7 9 10 11 -5 -3 -4 -1]

**NOTE:** Don't use functions and infinite loops

7. Program of Rotate and Delete.

Now we have problem with numbers. Take a list of numbers in which two types of operations allowed which is rotation and deletion. The process of doing these 2 operations are that first rotate the list in clockwise direction then delete the last element. In short rotate the list nth times and then deletes the nth last element. If the nth last element does not exists then delete the first element present in the list. So your task is to find out which is the last element that we delete from the list so that the list becomes empty after removing it.

For example, take list as

A = [1,2,3,4,5,6]

Rotate the list clockwise i.e. after rotation the list becomes A = [6,1,2,3,4,5] and delete the last element that is "5" so A = [6,1,2,3,4]. Again rotate the list for the second time and deletes the second last element that is "2" so A = [4,6,1,3], doing these steps when we reach 4th time, 4th last element does not exists so delete 1st element i.e., "1" so A={3,6}. So continuing this

procedure the last element in A is "3", so o/p will be 3.

Example:

Input

list = [1 2 3 4 5 6]

1. Rotate ==> [6, 1, 2, 3, 4, 5]
2. Delete (last) ==> [6, 1, 2, 3, 4]
3. Rotate ==> [4, 6, 1, 2, 3]
4. Delete (2nd last) ==> [4, 6, 1, 3]
5. Rotate ==> [3, 4, 6, 1]
6. Delete (3rd last) ==> [3, 6, 1]
7. Rotate ==> [1, 3, 6]
8. Delete (4th last) ==> No 4th Last, so delete 1st element ==> [3, 6]
9. Rotate ==> [6, 3]
10. Delete (5th last) ==> No 5th last element, so delete 1st element, 6, now list becomes ==> 3

Output:

3

Input

list = [1 2 3 4]

Rotate ==> [4 1 2 3]

Delete (last) ==> [4 1 2]

Rotate ==> [2 4 1]

Delete (2nd last) ==> [2 1]

Rotate ==> [1 2]

Delete (3rd last, no 3rd last, so delete 1st element 1) ==> Result 2

Output

2

**NOTE:**

1. Don't use functions and infinite loops
2. Code should be generic but can directly assign the list as required, say mylist = [1, 2, 3, 4, 5, 6]
3. The exception raised when trying to delete an element of an index which is not present is IndexError. Say, example mylist = [1, 3, 6] and trying to delete -4 element, which is not present. So, it would raise exception IndexError.

8. Write a "Python" program for demonstrating nested for loop for printing multiplication tables of mintablenum to maxtablenum, where mintablenum and maxtablenum are user entered numbers. The code should be generic, just giving an example for your understanding.

If user entered 2 and 6 then it should print multiplication tables from 2 to 6 having each table in one row only, which is as follows:

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

4 8 12 16 20 24 28 32 36 40  
5 10 15 20 25 30 35 40 45 50  
6 12 18 24 30 36 42 48 54 60

9. Write a program in Python to find the product of Primes. Given two numbers begin and end (both inclusive) find the product of primes within this range. If there are no primes in that range you must print 1.  
Print the message exactly as shown in samples, code should be generic, given samples for understanding purposes only.

Samples:

Enter Start Number: 1

Enter End Number: 10

The primes between 1 and 10 are [2,3,5,7] and their product is  $2*3*5*7=210$ .

Enter Start Number: 5

Enter End Number: 20

The primes between 5 and 20 are [5,7,11,13,17,19] and their product is 1616615.

**NOTE:**

1. Don't use functions and infinite loops.
2. The prime number check should be done with limit of square root of number but don't do from 2 till the number.

10. Write a program in Python for Special factorial number. Given a number say, num, the task is to find first natural number, x, whose factorial is divisible by num.

Input: 16

Output: 6

Explanation:

The required number is 6 whose factorial 720 is divisible by 16

Input: 5

Output: 5

Explanation:

The required number is 5 whose factorial 120 is divisible by 5

Input: 25

Output: 10

Explanation:

The required number is 10 whose factorial 3628800 is divisible by 25

11. Write a program in Python for Square Numbers (Don't write with functions (user defined) and infinite condition/loops).

A number k is called a square number if for some value of  $d > 1$ ,  $k \% (d*d) = 0$ .

Given a number N, the task is to find the total number of positive square numbers less than or equal to N.

Repeat the process until user enters 0 and ignore negative input.

4 -> (divisible by 2\*2)

8 ->(divisible by 2\*2)

9 ->(divisible by 3\*3)

Enter Number to find the square numbers: 2

No numbers less than or equal 2 are square numbers

Enter Number to find the square numbers: 4

Square numbers are: 4

Enter Number to find the square numbers: -10

Please enter +ve number only or 0 to exit

Enter Number to find the square numbers: 10

Square numbers are: 4, 8, 9

Enter Number to find the square numbers: 0

End of program

12. Given a series of numbers 3,10,21,36 ...., and series starting from  $n=1$ , find the pattern and output the  $n$ th value of above series. ( $1 \leq n \leq 100$ )

Example:

Input:

1  
2  
3  
4  
5  
6

Output:

3  
10  
21  
36  
55  
78

13. Given a non-negative integer num, repeatedly add all its digits until the result has only one digit. ( $1 \leq n \leq 10^9$ )

Input :

1  
98

Output :

1

## Functions

- Write a program in Python for calculating time difference as below.  
 Given the start time and end time of a race in the format hh:mm:ss. Your task is to print the time taken to complete the race in [hh:mm:ss]  
 You can initialize start time and end time in list. Implement logic in function by accepting start time and end time as arguments and return time difference. Store time difference in list as [hours, mins, secs]  
 Following is just an example, code should be generic. But can initialize starttime and endtime in list.  
  
 If, starttime = [13, 50, 45], endtime = [14, 55, 50]. The result should be, timediff = [1,5,5]  
 If, starttime = [13, 50, 45], endtime = [14, 45, 20]. The result should be, timediff = [0,54,35]  
 If, starttime = [13, 50, 45], endtime = [15, 00, 20]. The result should be, timediff = [1,9,35]
- Write a program in Python to perform the following:
  - Accept a list or accept numbers and form into list or initialize list with few elements and name it as mylist. Accept a number from user say, power.
  - Pass this mylist and power to function called elements\_power(). If the power is within the range of 1 to 5 it is ok, otherwise take it as 5. Assess each element of list and raise its element to the power. Return nothing from function.
  - Print the list in main program.
 Ex: Just a sample code to be generic,  
 If mylist is [1, 2, 3, 4, 5] and power is 3 then mylist would be printed in main as [1, 8, 27, 64, 125]
- Write a program in Python to perform product of complex numbers. Accept input as follows:
  - Accept first list from user, say list1
  - Accept second list from user, say list2
  - Both the lists (list1 and list2) to contain equal number of elements. Otherwise, throw an error and stop processing.
  - Pass these lists to function (say complex\_numbers\_mult(list1, list2)). The function need to perform multiplication and return result.
  - Main program to print the result along with complex numbers with multiplication symbols as in input/output section.

Input/Output:

Sample 1:  
 Enter List 1:[1, 2, 3, 4, 5]  
 Enter List 2:[1, 2, 3, 4, 5]  
 Complex numbers multiplication of (1+1j)\*(2+2j)\*(3+3j)\*(4+4j)\*(5+5j) is (-480-480j)

Sample 2:  
 Enter List 1:[1, 2, 3, 4, 5]  
 Enter List 2:[0, 0, 0, 0, 0]  
 Complex numbers multiplication of (1+0j)\*(2+0j)\*(3+0j)\*(4+0j)\*(5+0j) is (120+0j)

Explanation:

1) We are to multiply the 5 complex numbers and so  $(1+0i)*(2+0i)*(3+0i)*(4+0i)*(5+0i)$  we can see that answer is  $120 + 0i$ .

2) Similarly for 2nd Testcase  $(1+1i)*(2+2i)*(3+3i)*(4+4i)*(5+5i) = -480 -480i$

4. Write a program in Python to build the smallest number from user entered string. Given a number N and string S of digits denoting a positive integer [ $N < \text{sizeof}(S)$ ], build the lowest number possible by removing N digits from S, without changing their order. Implement the functionality in function by accepting string and number and return the smallest formed number.

NOTE: N should be less than length of (S).

Input

Enter Number of Digits: 4

Enter String: "1234567"

Output

The smallest number from "1234567" is 123

Input

Enter Number of Digits: 3

Enter String: "3435335"

Output

The smallest number from "3435335" is 3333

Input

Enter Number of Digits: 2

Enter String: "1034"

Output

The smallest number from "1034" is 03 (or 3)

5. 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

6. 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
```

7. 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 accept 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.

8. 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.