

1. Write a Python function that takes a sequence of numbers and determines whether all the numbers are different from each other.
2. Write a Python program to create all possible strings by using 'a', 'e', 'i', 'o', 'u'. Use the characters exactly once.
3. Write a Python program to remove and print every third number from a list of numbers until the list becomes empty.
4. Write a Python program to find unique triplets whose three elements gives the sum of zero from an array of n integers.
5. Write a Python program to create the combinations of 3 digit combo.
6. Write a Python program to print a long text, convert the string to a list and print all the words and their frequencies.
7. Write a Python program to count the number of each character of a given text of a text file.
8. Write a Python program to get the top stories from Google news.
9. Write a Python program to get a list of locally installed Python modules.
10. Write a Python program to display some information about the OS where the script is running.
11. Write a Python program to check the sum of three elements (each from an array) from three arrays is equal to a target value. Print all those three-element combinations.

Sample data:

```
/*  
X = [10, 20, 20, 20]  
Y = [10, 20, 30, 40]  
Z = [10, 30, 40, 20]  
target = 70  
*/
```

12. Write a Python program to create all possible permutations from a given collection of distinct numbers.

13. Write a Python program to get all possible two digit letter combinations from a digit (1 to 9) string.

```
string_maps = {  
"1": "abc",  
"2": "def",  
"3": "ghi",  
"4": "jkl",  
"5": "mno",  
"6": "pqrs",  
"7": "tuv",  
"8": "wxy",  
"9": "z"  
}
```

14. Write a Python program to add two positive integers without using the '+' operator.

Note: Use bit wise operations to add two numbers.

15. Write a Python program to check the priority of the four operators (+, -, *, /).

16. Write a Python program to get the third side of right angled triangle from two given sides.

17. Write a Python program to get all strobogrammatic numbers that are of length n.

A strobogrammatic number is a number whose numeral is rotationally symmetric, so that it appears the same when rotated 180 degrees. In other words, the numeral looks the same right-side up and upside down (e.g., 69, 96, 1001).

For example,

Given n = 2, return ["11", "69", "88", "96"].

Given n = 3, return ['818', '111', '916', '619', '808', '101', '906', '609', '888', '181', '986', '689']

18. Write a Python program to find the median among three given numbers.

19. Write a Python program to find the value of n where n degrees of number 2 are written sequentially in a line without spaces.

20. Write a Python program to find the number of zeros at the end of a factorial of a given positive number.

Range of the number(n): (1 <= n <= 2*10⁹).

21. Write a Python program to find the number of notes (Sample of notes: 10, 20, 50, 100, 200 and 500) against a given amount.

Range - Number of notes(n) : n (1 <= n <= 1000000)

22. Write a Python program to create a sequence where the first four members of the sequence are equal to one, and each successive term of the sequence is equal to the sum of the four previous ones. Find the Nth member of the sequence.

23. Write a Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive.

24. Write a Python program to find the number of divisors of a given integer is even or odd.

25. Write a Python program to find the digits which are absent in a given mobile number.

26. Write a Python program to compute the summation of the absolute difference of all distinct pairs in a given array (non-decreasing order).

Sample array: [1, 2, 3]

Then all the distinct pairs will be:

1 2

1 3

2 3

27. Write a Python program to find the type of the progression (arithmetic progression/geometric progression) and the next successive member of a given three successive members of a sequence. According to Wikipedia, an arithmetic progression (AP) is a sequence of numbers such that the difference of any two successive members of the sequence is a constant. For instance, the sequence 3, 5, 7, 9, 11, 13, . . . is an arithmetic progression with common difference 2. For this problem, we will limit ourselves to arithmetic progression whose common difference is a non-zero integer. On the other hand, a geometric progression (GP) is a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed non-zero number called the common ratio. For example, the sequence 2, 6, 18, 54, . . . is a geometric progression with common ratio 3. For this problem, we will limit ourselves to geometric progression whose common ratio is a non-zero integer.

28. Write a Python program to print the length of the series and the series from the given 3rd term, 3rd last term and the sum of a series.

Sample Data:

Input third term of the series: 3

Input 3rd last term: 3

Input Sum of the series: 15

Length of the series: 5

Series:

1 2 3 4 5

29. Write a Python program to find common divisors between two numbers in a given pair.

30. Write a Python program to reverse the digits of a given number and add it to the original, If the sum is not a palindrome repeat this procedure.

Note: A palindrome is a word, number, or other sequence of characters which reads the same backward as forward, such as madam or racecar.

31. Write a Python program to count the number of carry operations for each of a set of addition problems.

According to Wikipedia " In elementary arithmetic, a carry is a digit that is transferred from one column of digits to another column of more significant digits. It is part of the standard algorithm to add numbers together by starting with the rightmost digits and working to the left. For example, when 6 and 7 are added to make 13, the "3" is written to the same column and the "1" is carried to the left".

32. Write a python program to find heights of the top three building in descending order from eight given buildings.

Input:

0 <= height of building (integer) <= 10,000

Input the heights of eight buildings:

25

35

15

16

30

45

37

39

Heights of the top three buildings:

45

39

37

33. Write a Python program to compute the digit number of sum of two given integers.

Input:

Each test case consists of two non-negative integers x and y which are separated by a space in a line.

0 <= x, y <= 1,000,000

Input two integers(a b):

5 7

Sum of two integers a and b.:

2

34. Write a Python program to check whether three given lengths (integers) of three sides form a right triangle. Print "Yes" if the given sides form a right triangle otherwise print "No".

Input:

Integers separated by a single space.

1 <= length of the side <= 1,000

Input three integers(sides of a triangle)

8 6 7

No

35. Write a Python program which solve the equation:

$$ax+by=c$$

$$dx+ey=f$$

Print the values of x, y where a, b, c, d, e and f are given.

Input:

a,b,c,d,e,f separated by a single space.

(-1,000 <= a,b,c,d,e,f <= 1,000)

Input the value of a, b, c, d, e, f:

5 8 6 7 9 4

Values of x and y:

-2.000 2.000

36. Write a Python program to compute the amount of the debt in n months. The borrowing amount is \$100,000 and the loan adds 5% interest of the debt and rounds it to the nearest 1,000 above month by month.

Input:

An integer n (0 <= n <= 100)

Input number of months: 7

Amount of debt: \$144000

37. Write a Python program which reads an integer n and find the number of combinations of a,b,c and d (0 <= a,b,c,d <= 9) where (a + b + c + d) will be equal to n.

Input:

n (1 <= n <= 50)

Input the number(n): 15

Number of combinations: 592

38. Write a Python program to print the number of prime numbers which are less than or equal to a given integer.

Input:

n (1 <= n <= 999,999)

Input the number(n): 35

Number of prime numbers which are less than or equal to n.: 11

39. Write a program to compute the radius and the central coordinate (x, y) of a circle which is constructed by three given points on the plane surface.

Input:

x1, y1, x2, y2, x3, y3 separated by a single space.

Input three coordinate of the circle:

9 3 6 8 3 6

Radius of the said circle:

3.358

Central coordinate (x, y) of the circle:

6.071 4.643

40. Write a Python program to check whether a point (x,y) is in a triangle or not. There is a triangle formed by three points.

Input:

x1,y1,x2,y2,x3,y3,yp separated by a single space.

Input three coordinate of the circle:

9 3 6 8 3 6

Radius of the said circle:

3.358

Central coordinate (x, y) of the circle:

6.071 4.643

41. Write a Python program to compute and print sum of two given integers (more than or equal to zero). If given integers or the sum have more than 80 digits, print "overflow".

Input first integer:

25

Input second integer:

22

Sum of the two integers: 47

42. Write a Python program that accepts six numbers as input and sorts them in descending order.

Input:

Input consists of six numbers n1, n2, n3, n4, n5, n6 (-100000 <= n1, n2, n3, n4, n5, n6 <= 100000).

The six numbers are separated by a space.

Input six integers:

15 30 25 14 35 40

After sorting the said integers:

40 35 30 25 15 14

43. Write a Python program to test whether two lines PQ and RS are parallel. The four points are P(x1, y1), Q(x2, y2), R(x3, y3), S(x4, y4).

Input:

x1,y1,x2,y2,x3,y3,yp separated by a single space

Input x1,y1,x2,y2,x3,y3,yp:

2 5 6 4 8 3 9 7

PQ and RS are not parallel

44. Write a Python program to find the maximum sum of a contiguous subsequence from a given sequence of numbers $a_1, a_2, a_3, \dots, a_n$. A subsequence of one element is also a continuous subsequence.

Input:

You can assume that $1 \leq n \leq 5000$ and $-100000 \leq a_i \leq 100000$.

Input numbers are separated by a space.

Input 0 to exit.

Input number of sequence of numbers you want to input (0 to exit):

3

Input numbers:

2

4

6

Maximum sum of the said contiguous subsequence: 12

Input number of sequence of numbers you want to input (0 to exit):

0

45. There are two circles C1 with radius r_1 , central coordinate (x_1, y_1) and C2 with radius r_2 and central coordinate (x_2, y_2) .

Write a Python program to test the followings -

- "C2 is in C1" if C2 is in C1
- "C1 is in C2" if C1 is in C2
- "Circumference of C1 and C2 intersect" if circumference of C1 and C2 intersect
- "C1 and C2 do not overlap" if C1 and C2 do not overlap and
- "Circumference of C1 and C2 will touch" if C1 and C2 touch

Input:

Input numbers (real numbers) are separated by a space.

Input $x_1, y_1, r_1, x_2, y_2, r_2$:

5 4 2 3 9 2

C1 and C2 do not overlap

Input $x_1, y_1, r_1, x_2, y_2, r_2$:

5 4 3 5 10 3

Circumference of C1 and C2 will touch

Input $x_1, y_1, r_1, x_2, y_2, r_2$:

6 4 3 10 4 2

Circumference of C1 and C2 intersect

Input $x_1, y_1, r_1, x_2, y_2, r_2$:

5 4 3 5 4 2

C2 is in C1

Input $x_1, y_1, r_1, x_2, y_2, r_2$:

5 4 2 5 4 3

C1 is in C2

46. Write a Python program to that reads a date (from 2016/1/1 to 2016/12/31) and prints the day of the date. Jan. 1, 2016, is Friday. Note that 2016 is a leap year.

Input:

Two integers m and d separated by a single space in a line, m ,d represent the month and the day.

Input month and date (separated by a single space):

5 15

Name of the date: Sunday

47. Write a Python program which reads a text (only alphabetical characters and spaces.) and prints two words. The first one is the word which is arise most frequently in the text. The second one is the word which has the maximum number of letters.

Note: A word is a sequence of letters which is separated by the spaces.

Input:

A text is given in a line with following condition:

- a. The number of letters in the text is less than or equal to 1000.
- b. The number of letters in a word is less than or equal to 32.
- c. There is only one word which is arise most frequently in given text.
- d. There is only one word which has the maximum number of letters in given text.

Input text: Thank you for your comment and your participation.

Output: your participation.

48. Write a Python program that reads n digits (given) chosen from 0 to 9 and prints the number of combinations where the sum of the digits equals to another given number (s). Do not use the same digits in a combination.

Input:

Two integers as number of combinations and their sum by a single space in a line. Input 0 0 to exit.

Input number of combinations and sum, input 0 0 to exit:

5 6

2 4

0 0

2

49. Write a Python program which reads the two adjoined sides and the diagonal of a parallelogram and check whether the parallelogram is a rectangle or a rhombus.

According to Wikipedia - parallelograms: In Euclidean geometry, a parallelogram is a simple (non-self-intersecting) quadrilateral with two pairs of parallel sides. The opposite or facing sides of a parallelogram are of equal length and the opposite angles of a parallelogram are of equal measure.

rectangles: In Euclidean plane geometry, a rectangle is a quadrilateral with four right angles. It can also be defined as an equiangular quadrilateral, since equiangular means that all of its angles are equal ($360^\circ/4 = 90^\circ$). It can also be defined as a parallelogram containing a right angle.

rhombus: In plane Euclidean geometry, a rhombus (plural rhombi or rhombuses) is a simple (non-self-intersecting) quadrilateral whose four sides all have the same length. Another name is equilateral quadrilateral, since equilateral means that all of its sides are equal in length. The rhombus is often called a diamond, after the diamonds suit in playing cards which resembles the projection of an octahedral diamond, or a lozenge, though the former sometimes refers specifically to a rhombus with a 60° angle, and the latter sometimes refers specifically to a rhombus with a 45° angle.

Input:

Two adjoined sides and the diagonal.

1 <= ai, bi, ci <= 1000, ai + bi > ci

Input two adjoined sides and the diagonal of a parallelogram (comma separated):

3,4,5

This is a rectangle.

50. Write a Python program to replace a string "Python" with "Java" and "Java" with "Python" in a given string.

Input:

English letters (including single byte alphanumeric characters, blanks, symbols) are given on one line. The length of the input character string is 1000 or less.

Input a text with two words 'Python' and 'Java'

Python is popular than Java

Java is popular than Python

51. Write a Python program to find the difference between the largest integer and the smallest integer which are created by 8 numbers from 0 to 9. The number that can be rearranged shall start with 0 as in 00135668.

Input:

Input an integer created by 8 numbers from 0 to 9.:

2345

Difference between the largest and the smallest integer from the given integer:

3087

52. Write a Python program to compute the sum of first n given prime numbers.

Input:

n (n <= 10000). Input 0 to exit the program.

Input a number (n<=10000) to compute the sum:(0 to exit)

25

Sum of first 25 prime numbers:

1060

53. Write a Python program that accept an even number (≥ 4 , Goldbach number) from the user and create a combinations that express the given number as a sum of two prime numbers. Print the number of combinations.

Goldbach number: A Goldbach number is a positive even integer that can be expressed as the sum of two odd primes.[4] Since four is the only even number greater than two that requires the even prime 2 in order to be written as the sum of two primes, another form of the statement of Goldbach's conjecture is that all even integers greater than 4 are Goldbach numbers.

The expression of a given even number as a sum of two primes is called a Goldbach partition of that number. The following are examples of Goldbach partitions for some even numbers:

$$6 = 3 + 3$$

$$8 = 3 + 5$$

$$10 = 3 + 7 = 5 + 5$$

$$12 = 7 + 5$$

...

$$100 = 3 + 97 = 11 + 89 = 17 + 83 = 29 + 71 = 41 + 59 = 47 + 53$$

Input an even number (0 to exit):

100

Number of combinations:

6

54. if you draw a straight line on a plane, the plane is divided into two regions. For example, if you pull two straight lines in parallel, you get three areas, and if you draw vertically one to the other you get 4 areas.

Write a Python program to create maximum number of regions obtained by drawing n given straight lines.

Input:

($1 \leq n \leq 10,000$)

Input number of straight lines (0 to exit):

5

Number of regions:

16

55. There are four different points on a plane, $P(x_p, y_p)$, $Q(x_q, y_q)$, $R(x_r, y_r)$ and $S(x_s, y_s)$. Write a Python program to test AB and CD are orthogonal or not.

Input:

$x_p, y_p, x_q, y_q, x_r, y_r, x_s$ and y_s are -100 to 100 respectively and each value can be up to 5 digits after the decimal point It is given as a real number including the number of. Output:

Output AB and CD are not orthogonal! or AB and CD are orthogonal!.

56. Write a Python program to sum of all numerical values (positive integers) embedded in a sentence.

Input:

Sentences with positive integers are given over multiple lines. Each line is a character string containing one-byte alphanumeric characters, symbols, spaces, or an empty line. However the input is 80 characters or less per line and the sum is 10,000 or less.

Input some text and numeric values (

Sum of the numeric values: 80

None

Input some text and numeric values (

Sum of the numeric values: 17

None

Input some text and numeric values (

Sum of the numeric values: 10

None

57. There are 10 vertical and horizontal squares on a plane. Each square is painted blue and green. Blue represents the sea, and green represents the land. When two green squares are in contact with the top and bottom, or right and left, they are said to be ground. The area created by only one green square is called "island". For example, there are five islands in the figure below.

Write a Python program to read the mass data and find the number of islands.

Input:

Input 10 rows of 10 numbers representing green squares (island) as 1 and blue squares (sea) as zeros

1100000111

1000000111

0000000111

0010001000

0000011100

0000111110

0001111111

1000111110

1100011100

1110001000

Number of islands:

5

58. When character are consecutive in a string , it is possible to shorten the character string by replacing the character with a certain rule. For example, in the case of the character string YYYYYY, if it is expressed as # 5 Y, it is compressed by one character.

Write a Python program to restore the original string by entering the compressed string with this rule. However, the # character does not appear in the restored character string.

Input:

The restored character string for each character on one line.

Original text: XY#6Z1#4023

XYZZZZZZ1000023

Original text: #39+1=1#30

999+1=1000

59. A convex polygon is a simple polygon in which no line segment between two points on the boundary ever goes outside the polygon. Equivalently, it is a simple polygon whose interior is a convex set. In a convex polygon, all interior angles are less than or equal to 180 degrees, while in a strictly convex polygon all interior angles are strictly less than 180 degrees.

Write a Python program that compute the area of the polygon . The vertices have the names vertex 1, vertex 2, vertex 3, ... vertex n according to the order of edge connections [Go to the editor](#)

Note: The original sentences are uppercase letters, lowercase letters, numbers, symbols, less than 100 letters, and consecutive letters are not more than 9 letters.

Input:

Input number of sides: 5

Side: 1

Input the Coordinate:

Input Coordinate x: 1

Input Coordinate y: 0

Side: 2

Input the Coordinate:

Input Coordinate x: 0

Input Coordinate y: 0

Side: 3

Input the Coordinate:

Input Coordinate x: 1

Input Coordinate y: 1

Side: 4

Input the Coordinate:

Input Coordinate x: 2

Input Coordinate y: 0

Side: 5

Input the Coordinate:

Input Coordinate x: -1

Input Coordinate y: 1

Area of the Polygon: 0.5

60. Internet search engine giant, such as Google accepts web pages around the world and classify them, creating a huge database. The search engines also analyze the search keywords entered by the user and create inquiries for database search. In both cases, complicated processing is carried out in order to realize efficient retrieval, but basics are all cutting out words from sentences.

Write a Python program to cut out words of 3 to 6 characters length from a given sentence not more than 1024 characters.

Input:

English sentences consisting of delimiters and alphanumeric characters are given on one line.

Input a sentence (1024 characters. max.)

The quick brown fox

3 to 6 characters length of words:

The quick brown fox

61. Arrange integers (0 to 99) as narrow hilltop, as illustrated in Figure 1. Reading such data representing huge, when starting from the top and proceeding according to the next rule to the bottom. Write a Python program that compute the maximum value of the sum of the passing integers. [Go to the editor](#)

Input:

A series of integers separated by commas are given in diamonds. No spaces are included in each line. The input example corresponds to Figure 1. The number of lines of data is less than 100 lines.

Output:

The maximum value of the sum of integers passing according to the rule on one line.

Input the numbers (ctrl+d to exit):

8

4, 9

9, 2, 1

3, 8, 5, 5

5, 6, 3, 7, 6

3, 8, 5, 5

9, 2, 1

4, 9

8

Maximum value of the sum of integers passing according to the rule on one line.

64

62. Write a Python program to find the number of combinations that satisfy $p + q + r + s = n$ where n is a given number ≤ 4000 and p, q, r, s in the range of 0 to 1000.

Input a positive integer: (ctrl+d to exit)

252

Number of combinations of a,b,c,d: 2731135

63. Write a Python program which adds up columns and rows of given table as shown in the specified figure.

Input number of rows/columns (0 to exit)

4

Input cell value:

25 69 51 26

68 35 29 54

54 57 45 63

61 68 47 59

Result:

25 69 51 26 171

68 35 29 54 186

54 57 45 63 219

61 68 47 59 235

208 229 172 202 811

Input number of rows/columns (0 to exit)

64. Given a list of numbers and a number k, write a Python program to check whether the sum of any two numbers from the list is equal to k or not.

For example, given [1, 5, 11, 5] and k = 16, return true since 11 + 5 is 16.

Sample Input:

([12, 5, 0, 5], 10)

([20, 20, 4, 5], 40)

([1, -1], 0)

([1, 1, 0], 0)

Sample Output:

True

True

True

False

65. In mathematics, a subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. For example, the sequence (A,B,D) is a subsequence of (A,B,C,D,E,F) obtained after removal of elements C, E, and F. The relation of one sequence being the subsequence of another is a preorder.

The subsequence should not be confused with substring (A,B,C,D) which can be derived from the above string (A,B,C,D,E,F) by deleting substring (E,F). The substring is a refinement of the subsequence.

The list of all subsequences for the word "apple" would be "a", "ap", "al", "ae", "app", "apl", "ape", "ale", "appl", "appe", "apple", "p", "pp", "pl", "pe", "ppl", "ppe", "ple", "pple", "l", "le", "e", "".

Write a Python program to find the longest word in set of words which is a subsequence of a given string.

Sample Input:

```
("Green", {"Gn", "Gren", "ree", "en"})
```

```
("pythonexercises", {"py", "ex", "exercises"})
```

Sample Output:

```
Gren
```

```
exercises
```

66. From Wikipedia, the free encyclopaedia:

A happy number is defined by the following process:

Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers.

Write a Python program to check whether a number is "happy" or not.

Sample Input:

```
(7)
```

```
(932)
```

```
(6)
```

Sample Output:

```
True
```

```
True
```

```
False
```

67. From Wikipedia,

A happy number is defined by the following process:

Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers.

Write a Python program to find and print the first 10 happy numbers.

Sample Input:

```
[:10]
```

Sample Output:

```
[1, 7, 10, 13, 19, 23, 28, 31, 32, 44]
```

68. Write a Python program to count the number of prime numbers less than a given non-negative number.

Sample Input:

(10)

(100)

Sample Output:

4

25

69. In abstract algebra, a group isomorphism is a function between two groups that sets up a one-to-one correspondence between the elements of the groups in a way that respects the given group operations. If there exists an isomorphism between two groups, then the groups are called isomorphic.

Two strings are isomorphic if the characters in string A can be replaced to get string B

Given "foo", "bar", return false.

Given "paper", "title", return true.

Write a Python program to check if two given strings are isomorphic to each other or not.

Sample Input:

("foo", "bar")

("bar", "foo")

("paper", "title")

("title", "paper")

("apple", "orange")

("aa", "ab")

("ab", "aa")

Sample Output:

False

False

True

True

False

False

False

70. Write a Python program to find the longest common prefix string amongst a given array of strings. Return false If there is no common prefix.

For Example, longest common prefix of "abcdefgh" and "abcefg" is "abc".

Sample Input:

["abcdefgh","abcefg"]

["w3r","w3resource"]

["Python","PHP", "Perl"]

["Python","PHP", "Java"]

Sample Output:

abc

w3r

P

71. Write a Python program to reverse only the vowels of a given string.

Sample Input:

("w3resource")

("Python")

("Perl")

("USA")

Sample Output:

w3resuorce

Python

Perl

ASU

72. Write a Python program to check whether a given integer is a palindrome or not.

Note: An integer is a palindrome when it reads the same backward as forward. Negative numbers are not palindromic.

Sample Input:

(100)

(252)

(-838)

Sample Output:

False

True

False

73. Write a Python program to remove the duplicate elements of a given array of numbers such that each element appear only once and return the new length of the given array.

Sample Input:

[0,0,1,1,2,2,3,3,4,4,4]

[1, 2, 2, 3, 4, 4]

Sample Output:

5

4

74. Write a Python program to calculate the maximum profit from selling and buying values of stock. An array of numbers represent the stock prices in chronological order.

For example, given [8, 10, 7, 5, 7, 15], the function will return 10, since the buying value of the stock is 5 dollars and sell value is 15 dollars.

Sample Input:

([8, 10, 7, 5, 7, 15])

([1, 2, 8, 1])

([])

Sample Output:

10

7

0

75. Write a Python program to remove all instances of a given value from a given array of integers and find the length of the new array.

Sample Input:

([1, 2, 3, 4, 5, 6, 7, 5], 5)

([10,10,10,10,10], 10)

([10,10,10,10,10], 20)

([], 1)

Sample Output:

6

0

5

0

76. Write a Python program to find the starting and ending position of a given value in a given array of integers, sorted in ascending order.

If the target is not found in the array, return [0, 0].

Input: [5, 7, 7, 8, 8, 8] target value = 8

Output: [0, 5]

Input: [1, 3, 6, 9, 13, 14] target value = 4

Output: [0, 0]

77. The price of a given stock on each day is stored in an array.

Write a Python program to find the maximum profit in one transaction i.e., buy one and sell one share of the stock from the given price value of the said array. You cannot sell a stock before you buy one.

Input (Stock price of each day): [224, 236, 247, 258, 259, 225]

Output: 35

Explanation:

236 - 224 = 12

247 - 224 = 23

258 - 224 = 34

259 - 224 = 35

225 - 224 = 1

247 - 236 = 11

258 - 236 = 22

259 - 236 = 23

225 - 236 = -11

258 - 247 = 11

259 - 247 = 12

225 - 247 = -22

259 - 258 = 1

225 - 258 = -33

225 - 259 = -34

78. Write a Python program to print a given N by M matrix of numbers line by line in forward > backwards > forward >... order.

Input matrix:

```
[[1, 2, 3,4],  
[5, 6, 7, 8],  
[0, 6, 2, 8],  
[2, 3, 0, 2]]
```

Output:

```
1  
2  
3  
4  
8  
7  
6  
5  
0  
6  
2  
8  
2  
0  
3  
2
```

79. Write a Python program to compute the largest product of three integers from a given list of integers.

Sample Input:

```
[-10, -20, 20, 1]  
[-1, -1, 4, 2, 1]  
[1, 2, 3, 4, 5, 6]
```

Sample Output:

```
4000  
8  
120
```

80. Write a Python program to find the first missing positive integer that does not exist in a given list.

Sample Input:

```
[2, 3, 7, 6, 8, -1, -10, 15, 16]  
[1, 2, 4, -7, 6, 8, 1, -10, 15]  
[1, 2, 3, 4, 5, 6, 7]  
[-2, -3, -1, 1, 2, 3]
```

Sample Output:

```
4  
3  
8  
4
```

81. Write a Python program to randomly generate a list with 10 even numbers between 1 and 100 inclusive.

Note: Use `random.sample()` to generate a list of random values.

Sample Input:

(1,100)

Sample Output:

[4, 22, 8, 20, 24, 12, 30, 98, 28, 48]

82. Write a Python program to calculate the median from a list of numbers.

Sample Input:

[1,2,3,4,5]

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

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

[1.0,2.11,3.3,4.2,5.22,6.55]

[1.0,2.11,3.3,4.2,5.22]

[2.0,12.11,22.3,24.12,55.22]

Sample Output:

3

3.5

3.5

3.75

3.3

22.3

83. Write a Python program to test whether a given number is symmetrical or not.

A number is symmetrical when it is equal of its reverse.

Sample Input:

(121)

(0)

(122)

(990099)

Sample Output:

True

True

False

True

84. Write a Python program that accepts a list of numbers. Count the negative numbers and compute the sum of the positive numbers of the said list. Return these values through a list.

Original list: [1, 2, 3, 4, 5]

Number of negative of numbers and sum of the positive numbers of the said list: [0, 15]

Original list: [-1, -2, -3, -4, -5]

[5, 0]

Number of negative of numbers and sum of the positive numbers of the said list: [5, 0]

Original list: [1, 2, 3, -4, -5]

[2, 6]

Number of negative of numbers and sum of the positive numbers of the said list: [2, 6]

Original list: [1, 2, -3, -4, -5]

[3, 3]

Number of negative of numbers and sum of the positive numbers of the said list: [3, 3]

85. From Wikipedia:

An isogram (also known as a "nonpattern word") is a logological term for a word or phrase without a repeating letter. It is also used by some people to mean a word or phrase in which each letter appears the same number of times, not necessarily just once. Conveniently, the word itself is an isogram in both senses of the word, making it autological.

Write a Python program to check whether a given string is an "isogram" or not.

Sample Input:

("w3resource")

("w3r")

("Python")

("Java")

Sample Output:

False

True

True

False

86. Write a Python program to count the number of equal numbers from three given integers.

Sample Input:

(1, 1, 1)

(1, 2, 2)

(-1, -2, -3)

(-1, -1, -1)

Sample Output:

3

2

0

3

87. Write a Python program to check whether a given employee code is exactly 8 digits or 12 digits. Return True if the employee code is valid and False if it's not.

Sample Input:

```
('12345678')  
( '1234567j')  
( '12345678j')  
( '123456789123')  
( '123456abcdef')
```

Sample Output:

```
True  
False  
False  
True  
False
```

88. Write a Python program that accept two strings and test if the letters in the second string are present in the first string.

Sample Input:

```
["python", "ypth"]  
["python", "ypths"]  
["python", "ypthon"]  
["123456", "01234"]  
["123456", "1234"]
```

Sample Output:

```
True  
False  
True  
False  
True
```

89. Write a Python program to compute the sum of the three lowest positive numbers from a given list of numbers.

Original list of numbers: [10, 20, 30, 40, 50, 60, 7]

Sum of the three lowest positive numbers of the said array: 37

Original list of numbers: [1, 2, 3, 4, 5]

Sum of the three lowest positive numbers of the said array: 6

Original list of numbers: [0, 1, 2, 3, 4, 5]

Sum of the three lowest positive numbers of the said array: 6

90. Write a Python program to replace all but the last five characters of a given string with "*" and returns the new string.

Original String: kdi39323swe

new string: *****23swe

Original String: 12345abcdef

new string: *****bcdef

Original String: 12345

new string: 12345

91. Write a Python program to count the number of arguments in a given function. [Go to the editor](#)

Sample Input:

()

(1)

(1, 2)

(1, 2, 3)

(1, 2, 3, 4)

[1, 2, 3, 4]

Sample Output:

0

1

2

3

4

1

92. Write a Python program to compute cumulative sum of numbers of a given list.

Note: Cumulative sum = sum of itself + all previous numbers in the said list.

Sample Input:

[10, 20, 30, 40, 50, 60, 7]

[1, 2, 3, 4, 5]

[0, 1, 2, 3, 4, 5]

Sample Output:

[10, 30, 60, 100, 150, 210, 217]

[1, 3, 6, 10, 15]

[0, 1, 3, 6, 10, 15]

93. Write a Python program to find the middle character(s) of a given string. If the length of the string is even return the two middle characters. If the length of the string is odd, return the middle character.

Original string: Python

Middle character(s) of the said string: th

Original string: PHP

Middle character(s) of the said string: H

Original string: Java

Middle character(s) of the said string: av

94. Write a Python program to find the largest product of the pair of adjacent elements from a given list of integers.

Original list: [1, 2, 3, 4, 5, 6]

Largest product of the pair of adjacent elements of the said list: 30

Original list: [1, 2, 3, 4, 5]

Largest product of the pair of adjacent elements of the said list: 20

Original list: [2, 3]

Largest product of the pair of adjacent elements of the said list: 6

95. Write a Python program to check whether every even index contains an even number and every odd index contains odd number of a given list.

Original list of numbers: [2, 1, 4, 3, 6, 7, 6, 3]

Check whether every even index contains an even number and every odd index contains odd number of a given list:

True

Original list of numbers: [2, 1, 4, 3, 6, 7, 6, 4]

Check whether every even index contains an even number and every odd index contains odd number of a given list:

False

Original list of numbers: [2, 1, 4, 3, 6, 7, 6, 4]

Check whether every even index contains an even number and every odd index contains odd number of a given list:

True

96. Write a Python program to check whether a given number is a narcissistic number or not.

If you are a reader of Greek mythology, then you are probably familiar with Narcissus. He was a hunter of exceptional beauty that he died because he was unable to leave a pool after falling in love with his own reflection. That's why I keep myself away from pools these days (kidding).

In mathematics, he has kins by the name of narcissistic numbers - numbers that can't get enough of themselves. In particular, they are numbers that are the sum of their digits when raised to the power of the number of digits.

For example, 371 is a narcissistic number; it has three digits, and if we cube each digits $3^3 + 7^3 + 1^3$ the sum is 371. Other 3-digit narcissistic numbers are

$$153 = 1^3 + 5^3 + 3^3$$

$$370 = 3^3 + 7^3 + 0^3$$

$$407 = 4^3 + 0^3 + 7^3.$$

There are also 4-digit narcissistic numbers, some of which are 1634, 8208, 9474 since

$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

$$8208 = 8^4 + 2^4 + 0^4 + 8^4$$

$$9474 = 9^4 + 4^4 + 7^4 + 4^4$$

It has been proven that there are only 88 narcissistic numbers (in the decimal system) and that the largest of which is

115,132,219,018,763,992,565,095,597,973,971,522,401

has 39 digits.

Ref.: <https://bit.ly/2qNYxo2>

Sample Input:

(153)

(370)

(407)

(409)

(1634)

(8208)

(9474)

(9475)

Sample Output:

True

True

True

False

True

True

True

False

97. Write a Python program to find the highest and lowest number from a given string of space separated integers.

Original string: 1 4 5 77 9 0

Highest and lowest number of the said string: (77, 0)

Original string: -1 -4 -5 -77 -9 0

Highest and lowest number of the said string: (0, -77)

Original string: 0 0

Highest and lowest number of the said string: (0, 0)

98. Write a Python program to check whether a sequence of numbers has an increasing trend or not.

Sample Input:

[1,2,3,4]

[1,2,5,3,4]

[-1,-2,-3,-4]

[-4,-3,-2,-1]

[1,2,3,4,0]

Sample Output:

True

False

False

True

False

99. Write a Python program to find the position of the second occurrence of a given string in another given string. If there is no such string return -1.

Sample Input:

("The quick brown fox jumps over the lazy dog", "the")

("the quick brown fox jumps over the lazy dog", "the")

Sample Output:

-1

31

100. Write a Python program to compute the sum of all items of a given array of integers where each integer is multiplied by its index. Return 0 if there is no number.

Sample Input:

[1,2,3,4]

[-1,-2,-3,-4]

[]

Sample Output:

20

-20

0

101. Write a Python program to find the name of the oldest student from a given dictionary containing the names and ages of a group of students.

Sample Input:

```
{"Bernita Ahner": 12, "Kristie Marsico": 11, "Sara Pardee": 14, "Fallon Fabiano": 11, "Nidia Dominique": 15}
```

```
{"Nilda Woodside": 12, "Jackelyn Pineda": 12.2, "Sofia Park": 12.4, "Joannie Archibald": 12.6, "Becki Saunder": 12.7}
```

Sample Output:

Nidia Dominique

Becki Saunder

102. Write a Python program to create a new string with no duplicate consecutive letters from a given string.

Sample Input:

```
("PPYYTTHON")
```

```
("PPyyythonn")
```

```
("Java")
```

```
("PPPHHHPPP")
```

Sample Output:

PYTHON

Python

Java

PHP

103. Write a Python program to check whether two given lines are parallel or not.

Note: Parallel lines are two or more lines that never intersect. Parallel Lines are like railroad tracks that never intersect.

The General Form of the equation of a straight line is: $ax + by = c$

The said straight line is represented in a list as [a, b, c]

Example of two parallel lines:

$x + 4y = 10$ and $x + 4y = 14$

Sample Input:

```
([2,3,4], [2,3,8])
```

```
([2,3,4], [4,-3,8])
```

Sample Output:

True

False

104. Write a Python program to find a number in a given matrix, which is maximum in its column and minimum in its row.

Sample Input:

Original matrix: [[1, 2], [2, 3]]

Number in the said matrix which is maximum in its column and minimum in its row:

[2]

Original matrix: [[1, 2, 3], [3, 4, 5]]

Number in the said matrix which is maximum in its column and minimum in its row:

[3]

Original matrix: [[7, 5, 6], [3, 4, 4], [6, 5, 7]]

Number in the said matrix which is maximum in its column and minimum in its row:

[5]

105. Write a Python program to check whether a given sequence is linear, quadratic or cubic.

Sequences are sets of numbers that are connected in some way.

Linear sequence:

A number pattern which increases or decreases by the same amount each time is called a linear sequence. The amount it increases or decreases by is known as the common difference.

Quadratic sequence:

In quadratic sequence, the difference between each term increases, or decreases, at a constant rate.

Cubic sequence:

Sequences where the 3rd difference are known as cubic sequence.

Sample Output:

Original Sequence: [0, 2, 4, 6, 8, 10]

Check the said sequence is Linear, Quadratic or Cubic?

Linear Sequence

Original Sequence: [1, 4, 9, 16, 25]

Check the said sequence is Linear, Quadratic or Cubic?

Quadratic Sequence

Original Sequence: [0, 12, 10, 0, -12, -20]

Check the said sequence is Linear, Quadratic or Cubic?

Cubic Sequence

Original Sequence: [1, 2, 3, 4, 5]

Check the said sequence is Linear, Quadratic or Cubic?

Linear Sequence

106. Write a Python program to test whether a given integer is Pandigital number or not.

From Wikipedia,

In mathematics, a pandigital number is an integer that in a given base has among its significant digits each digit used in the base at least once.

For example,

1223334444555556666667777778888888999999990 is a pandigital number in base 10.

The first few pandigital base 10 numbers are given by:

1023456789, 1023456798, 1023456879, 1023456897, 1023456978, 1023456987, 1023457689

Sample Output:

Original number: 1023456897 Check the said number is Pandigital number or not? True Original

number: 1023456798 Check the said number is Pandigital number or not? True Original number:

1023457689 Check the said number is Pandigital number or not? True Original number:

1023456789 Check the said number is Pandigital number or not? True Original number: 102345679

Check the said number is Pandigital number or not? False

107. Write a Python program to check whether a given number is Oddish or Evenish.

A number is called "Oddish" if the sum of all of its digits is odd, and a number is called "Evenish" if the sum of all of its digits is even.

Sample Output:

Original Number 120

Check whether the sum of all digits of the said number is odd or even!

Oddish

Original Number 321

Check whether the sum of all digits of the said number is odd or even!

Evenish

Original Number 43

Check whether the sum of all digits of the said number is odd or even!

Oddish

Original Number 4433

Check whether the sum of all digits of the said number is odd or even!

Evenish

Original Number 373

Check whether the sum of all digits of the said number is odd or even!

Oddish

108. Write a Python program that takes three integers and check whether the sum of the last digit of first number and the last digit of second number equal to the last digit of third number.

Sample Input:

(12, 26, 44)

(145, 122, 1010)

(0, 20, 40)

(1, 22, 40)

(145, 129, 104)

Sample Output:

True

False

True

False

True

109. Write a Python program find the indices of all occurrences of a given item in a given list.

Sample Input:

([1,2,3,4,5,2], 2)

([3,1,2,3,4,5,6,3,3], 3)

([1,2,3,-4,5,2,-4], -4)

Sample Output:

Original list of numbers: [1, 2, 3, 4, 5, 2]

Given Number 2

Indices of all occurrences of the said item in the given list:

[1, 5]

Original list of numbers: [3, 1, 2, 3, 4, 5, 6, 3, 3]

Given Number 3

Indices of all occurrences of the said item in the given list:

[0, 3, 7, 8]

Original list of numbers: [1, 2, 3, -4, 5, 2, -4]

Given Number -4

Indices of all occurrences of the said item in the given list:

[3, 6]

Original list of numbers: [1, 2, 3, 4, 5, 2]

Given Number 7

Indices of all occurrences of the said item in the given list:

[]

110. Write a Python program to remove the duplicate numbers from a given list of numbers.

Sample Input:

([1,2,3,2,3,4,5])

([1,2,3,2,4,5])

([1,2,3,4,5])

Sample Output:

Original list of numbers: [1, 2, 3, 2, 3, 4, 5]

After removing the duplicate numbers from the said list:

[1, 4, 5]

Original list of numbers: [1, 2, 3, 2, 4, 5]

After removing the duplicate numbers from the said list:

[1, 3, 4, 5]

Original list of numbers: [1, 2, 3, 4, 5]

After removing the duplicate numbers from the said list:

[1, 2, 3, 4, 5]

111. Write a Python program to check whether two given circles (given center (x,y) and radius) are intersecting. Return true for intersecting otherwise false.

Sample Input:

([1,2, 4], [1,2, 8])

([0,0, 2], [10,10, 5])

Sample Output:

True

False

112. Write a Python program to compute the digit distance between two integers.

The digit distance between two numbers is the absolute value of the difference of those numbers.

For example, the distance between 3 and -3 on the number line given by the $|3 - (-3)| = |3 + 3| = 6$ units

Digit distance of 123 and 256 is

Since $|1 - 2| + |2 - 5| + |3 - 6| = 1 + 3 + 3 = 7$

Sample Input:

(123, 256)

(23, 56)

(1, 2)

(24232, 45645)

Sample Output:

7

6

1

11

113. Write a Python program to reverse all the words which have even length.

Sample Input:

```
("The quick brown fox jumps over the lazy dog")
```

```
("Python Exercises")
```

Sample Output:

```
The quick brown fox jumps revo the yzal dog
```

```
nohtyP Exercises
```

114. Write a Python program to print letters from the English alphabet from a-z and A-Z.

Sample Input:

```
("Alphabet from a-z:")
```

```
("\\nAlphabet from A-Z:")
```

Sample Output:

```
Alphabet from a-z:
```

```
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
```

```
Alphabet from A-Z:
```

```
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
```

115. Write a Python program to generate and prints a list of numbers from 1 to 10.

Sample Input:

```
range(1,10)
```

Sample Output:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
['1', '2', '3', '4', '5', '6', '7', '8', '9']
```

116. Write a Python program to identify nonprime numbers between 1 to 100 (integers). Print the nonprime numbers.

Sample Input:

```
range(1, 101)
```

Sample Output:

```
Nonprime numbers between 1 to 100:
```

```
4
```

```
6
```

```
8
```

```
9
```

```
10
```

```
..
```

```
96
```

```
98
```

```
99
```

```
100
```


117. Write a Python program to make a request to a web page, and test the status code, also display the html code of the specified web page.

Sample Output:

Web page status: <Response [200]>

HTML code of the above web page:

```
<!doctype html>
<html>
<head>
<title>Example Domain</title>
<meta charset="utf-8" />
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
</head>
<body>
<div>
<h1>Example Domain</h1>
<p>This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.</p>
<p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

118. In multiprocessing, processes are spawned by creating a Process object. Write a Python program to show the individual process IDs (parent process, process id etc.) involved.

Sample Output:

Main line

module name: __main__

parent process: 23967

process id: 27986

function f

module name: __main__

parent process: 27986

process id: 27987

hello bob

119. Write a Python program to check if two given numbers are Co Prime or not. Return True if two numbers are Co Prime otherwise return false.

Sample Input:

(17, 13)

(17, 21)

(15, 21)

(25, 45)

Sample Output:

True

True

False

False

120. Write a Python program to calculate Euclid's totient function of a given integer. Use a primitive method to calculate Euclid's totient function.

Sample Input:

(10)

(15)

(33)

Sample Output:

4

8

20

121. Write a Python program to create a coded string from a given string, using specified formula. Replace all 'P' with '9', 'T' with '0', 'S' with '1', 'H' with '6' and 'A' with '8'

Original string: PHP

Coded string: 969

Original string: JAVASCRIPT

Coded string: J8V81CRI90

122. Write a Python program to check if a given string contains only lowercase or uppercase characters.

Original string: PHP

Coded string: True

Original string: javascript

Coded string: True

Original string: JavaScript

Coded string: False

123. Write a Python program to remove the first and last elements from a given string.

Original string: PHP

Removing the first and last elements from the said string: H

Original string: Python

Removing the first and last elements from the said string: ytho

Original string: JavaScript

Removing the first and last elements from the said string: avaScrip

124. Write a Python program to check if a given string contains two similar consecutive letters.

Original string: PHP

Check for consecutive similar letters! False

Original string: PPHP

Check for consecutive similar letters! True

Original string: PHPP

Check for consecutive similar letters! True

125. Write a Python program to reverse a given string in lower case.

Original string: PHP

Reverse the said string in lower case: php

Original string: JavaScript

Reverse the said string in lower case: tpircsavaj

Original string: PHPP

Reverse the said string in lower case: pphp

126. Write a Python program to convert the letters of a given string (same case-upper/lower) into alphabetical order.

Original string: PHP

Convert the letters of the said string into alphabetical order: HPP

Original string: javascript

Convert the letters of the said string into alphabetical order: aacijprstv

Original string: python

Convert the letters of the said string into alphabetical order: hnopty

127. Write a Python program to check whether the average value of the elements of a given array of numbers is a whole number or not.

Original array:

1 3 5 7 9

Check the average value of the elements of the said array is a whole number or not: True

Original array:

2 4 2 6 4 8

Check the average value of the elements of the said array is a whole number or not:

False

128. Write a Python program to remove all vowels from a given string.

Original string: Python

After removing all the vowels from the said string: Pythn

Original string: C Sharp

After removing all the vowels from the said string: C Shrp

Original string: JavaScript

After removing all the vowels from the said string: JvScrip

129. Write a Python program to get the index number of all lower case letters in a given string.

Original string: Python

Indices of all lower case letters of the said string: [1, 2, 3, 4, 5] Original string: JavaScript

Indices of all lower case letters of the said string: [1, 2, 3, 5, 6, 7, 8, 9] Original string: PHP

Indices of all lower case letters of the said string: []

130. Write a Python program to check whether a given month and year contains a Monday 13th.

Month No.: 11 Year: 2022

Check whether the said month and year contains a Monday 13th.: False

Month No.: 6 Year: 2022

Check whether the said month and year contains a Monday 13th.: True

131. Write a Python program to count number of zeros and ones in the binary representation of a given integer.

Original number: 12

Number of ones and zeros in the binary representation of the said number: Number of zeros: 2,

Number of ones: 2

Original number: 1234

Number of ones and zeros in the binary representation of the said number: Number of zeros: 6,

Number of ones: 5

132. Write a Python program to find all the factors of a given natural number.

Factors:

The factors of a number are the numbers that divide into it exactly. The number 12 has six factors:

1, 2, 3, 4, 6 and 12 If 12 is divided by any of the six factors then the answer will be a whole number.

For example:

$12 / 3 = 4$

Original Number: 1

Factors of the said number: {1}

Original Number: 12

Factors of the said number: {1, 2, 3, 4, 6, 12}

Original Number: 100

Factors of the said number: {1, 2, 4, 100, 5, 10, 50, 20, 25}

133. Write a Python program to compute the sum of the negative and positive numbers of an array of integers and display the largest sum.

Original array elements: {0, 15, 16, 17, -14, -13, -12, -11, -10, 18, 19, 20}

Largest sum - Positive/Negative numbers of the said array: 105

Original array elements: {0, 3, 4, 5, 9, -22, -44, -11}

Largest sum - Positive/Negative numbers of the said array: -77

134. Write a Python program to alternate the case of each letter in a given string and the first letter of the said string must be uppercase.

Original string: Python Exercises

After alternating the case of each letter of the said string: PyThOn ExErCiSeS

Original string: C# is used to develop web apps, desktop apps, mobile apps, games and much more.

After alternating the case of each letter of the said string: C# iS uSeD tO dEvElOp WeB aPpS, dEsKtOp ApPs, MoBiLe ApPs, GaMeS aNd MuCh MoRe.

135. Write a Python program to get the Least Common Multiple (LCM) of more than two numbers. Take the numbers from a given list of positive integers.

From Wikipedia,

In arithmetic and number theory, the least common multiple, lowest common multiple, or smallest common multiple of two integers a and b , usually denoted by $\text{lcm}(a, b)$, is the smallest positive integer that is divisible by both a and b . Since division of integers by zero is undefined, this definition has meaning only if a and b are both different from zero. However, some authors define $\text{lcm}(a, 0)$ as 0 for all a , which is the result of taking the lcm to be the least upper bound in the lattice of divisibility.

Original list elements: [4, 6, 8]

LCM of the numbers of the said array of positive integers: 24

Original list elements: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

LCM of the numbers of the said array of positive integers: 2520

Original list elements: [48, 72, 108]

LCM of the numbers of the said array of positive integers: 432

136. Write a Python program to reverse all the words which have odd length.

Original string: The quick brown fox jumps over the lazy dog

Reverse all the words of the said string which have odd length: ehT kciuq nworb xof spmuj over eht lazy god

Original string: Python Exercises

Reverse all the words of the said string which have odd length: Python sesicrexE

137. Write a Python program to find the longest common ending between two given strings.

Original strings: running ruminating

Common ending between said two strings: ing

Original strings: thisisatest testing123testing

Common ending between said two strings:

138. Write a Python program to reverse the binary representation of an given integer and convert the reversed binary number into an integer.

Original number: 13

Reverse the binary representation of the said integer and convert it into an integer: 11

Original number: 145

Reverse the binary representation of the said integer and convert it into an integer: 137

Original number: 1342

Reverse the binary representation of the said integer and convert it into an integer: 997

139. Write a Python program to find the closest palindrome number of a given integer. If there are two palindrome numbers in absolute distance return the smaller number.

Original number: 120

Closest Palindrome number of the said number: 121

Original number: 321

Closest Palindrome number of the said number: 323

Original number: 43

Closest Palindrome number of the said number: 44

Original number: 1234

Closest Palindrome number of the said number: 1221

140. Write a Python program to convert all items in a given list to float values.

Original list:

```
['0.49', '0.54', '0.54', '0.54', '0.54', '0.54', '0.55', '0.54', '0.54', '0.54', '0.55', '0.55', '0.55', '0.54', '0.55', '0.55', '0.54', '0.55', '0.55', '0.54']
```

List of Floats:

```
[0.49, 0.54, 0.54, 0.54, 0.54, 0.54, 0.55, 0.54, 0.54, 0.54, 0.55, 0.55, 0.55, 0.54, 0.55, 0.55, 0.54, 0.55, 0.55, 0.54]
```

141. Write a Python program to get the domain name using PTR DNS records from a given IP address.

Domain name using PTR DNS:

dns.google

ec2-13-251-106-90.ap-southeast-1.compute.amazonaws.com

dns.google

ec2-23-23-212-126.compute-1.amazonaws.com

142. Write a Python program to check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones of same length in a given string. Return True/False. [Go to the editor](#)

Original sequence: 001011

Check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones in the said string:

False

Original sequence: 01010101

Check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones in the said string:

True

Original sequence: 00

Check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones in the said string:

False

Original sequence: 000111000111

Check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones in the said string:

True

Original sequence: 00011100011

Check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones in the said string:

False

Original sequence: 0011101

Check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones in the said string:

False

143. Write a Python program to print Emojis using unicode characters or CLDR (Common Locale Data Repository) short names.

```
Smiling face with heart-eyes:
😍
Unamused face:
😏
Beaming face with smiling eyes:
😄
Grinning face with sweat:
😅
Face with tears of joy:
😂
Slightly smiling face:
😊
Smiling face with halo:
😇
Zipper-mouth face:
😬
Grinning face:
😁
Rolling on the floor laughing:
😂
```

144. Write a Python program to convert integer to string.

Sample Input:

```
language = "Python"
```

```
version = 3.6
```

sample Output:

Language: Python, Version: 3.6

145. Write a Python program to find the largest and smallest digit of a given number.

Original Number: 9387422

Largest Digit of the said number: 9

Smallest Digit of the said number: 2

Original Number: 500

Largest Digit of the said number: 5

Smallest Digit of the said number: 0

Original Number: 231548

Largest Digit of the said number: 8

Smallest Digit of the said number: 1

146. A Python list contains two positive integers. Write a Python program to check whether the cube root of the first number is equal to the square root of the second number.

Sample Data:

([8, 4]) -> True

([64, 16]) -> True

([64, 36]) -> False

147. A Python list contains three positive integers. Write a Python program to check whether the sum of the digits in each number is equal or not. Return true otherwise false.

Sample Data:

([13, 4, 22]) -> True

([-13, 4, 22]) -> False

([45, 63, 90]) -> True

148. A Python list contains some positive integers. Write a Python program to count the numbers that are greater than the previous number on the list.

Sample Data:

([1, 4, 7, 9, 11, 5]) -> 4

([1, 3, 3, 2, 2]) -> 1

([4, 3, 2, 1]) -> 0

149. Write a Python program that takes a positive integer and create an N x N square filled with the integer N. Display the N x N square.

Sample Data:

(2) -> [[2, 2], [2, 2]]

(5) -> [[5, 5, 5, 5, 5], [5, 5, 5, 5, 5], [5, 5, 5, 5, 5], [5, 5, 5, 5, 5], [5, 5, 5, 5, 5]]

(-6) -> []

150. Write a Python program that takes a positive integer and calculate the cube root of the number until the number is less than three. Count the number of steps to complete the said process.

Sample Data:

(3) -> 1

(39) -> 2

(10000) -> 2