# CSD Domain Skills 3

1.Write a program to find the reverse factorial of a number.

2.Write a program to check whether a given sentence is a Pangram or not.

Ex: Pack my box with five dozen liquor jugs.

3.There are N towns in a coordinate plane. Town i is located at coordinate (xi,yi).The distance between town I and j is |xi-xj|+|yi-yj|. Your task is to compute the sum of the distance between each pair of towns.

Test case: (3,2), (4, -4), (5,0).

4. One Interval:

You are given a two-dimensional list of integers intervals where each list represents an inclusive [start, end] interval. For an interval [a, b] (where a < b), its size is defined as b-a. You must add one interval to the given list such that, after merging all the intervals, you get exactly one range left. Return the minimum possible size of the added interval.

Constraints:

0 ≤ n ≤ 100,000 where n is the length of intervals.

Example 1:

Input:

intervals = [ [10, 20], [25, 100]]

Output:

5

Explanation:

We can add the interval [20, 25] which is the smallest possible interval.

5.Given a string Str, which contains numbers (0 to 9) and also letters of the English alphabet (‘a’ to ‘z’ and ‘A’ to ‘Z’). The task is to reverse the string in such a way that the positions of numbers in the string are left unaltered.

Example 1:

Input: a1b2igh3

Output: h1g2iba3

Explanation: Without changing the positions of 1,2 and 3 only the positions of characters ‘h’,’g’,’i’’b’ and ‘a’ have been reversed.

Example 2:

Input: Ab5c7de96

Output: Ed5c7de96

Explanation: Without changing the positions of 5,7,9, and 6, only the positions of characters ‘A’,’b’,’c’,’d’ and ‘e’ have been reversed.

Constraints:

The input string should only hold letters of the alphabet and numbers.

Size of Str<=500

Input format for testing:

The candidate must write the code to accept a single string Str.

Output format for testing:

The output should be a string only. (See the output format in Example 1 and Example 2).

Additional messages in the output will result in the failure of test cases.

Instructions:

The system does not allow any kind of hard-coded input value/values.

The written program code by the candidate will be verified against the inputs which are

supplied from the system.

6.Palindrome Conversion

Write a program to identify whether the given number N is palindrome based upon the following operations:

-Add the given number and its reverse.

-Check whether the obtained sum after the first operation is a palindrome or not and if not then repeat the above operation.

The given operation will continue until a palindromic number is found. Print the resultant palindromic number as the output.

Note:

Palindrome: A sequence of letters/characters which reads the same backward and forwards.

A single letter/character is also considered a palindrome.

Input Format:

The input consists of a single line:

-The input contains a single integer denoting N.

The input will be read from the STDIN by the candidate.

Output Format:

Print the resultant palindromic number for the given input.

The output will be matched to the candidates output printed on the STDOUT.

Constraints

0<N<3\*109

Example:

Input:

195

Output:

9339

Explanation:

195 + 591 = 786(non-palindromic number)

786 + 687 = 1473(non-palindromic number)

1473 + 3741 = 5214(non-palindromic number)

5214 + 4125 = 9339(Palindromic number)

Sample Input:

100

Sample Output:

101

Instructions:

-Program should take input from standard input and print output to standard output.

-Your code is judged by an automated system, do not write any additional welcome/greeting messages.

- “Save and Test” only checks for basic test cases, more rigorous cases will be used to judge your code.

-Additional score will be given for writing optimized code both in terms of memory and execution time.

7. Bleak Number:

In Mathematics, each number has one special number which it supports. It is chosen as follows:

It counts the number of ones in its own binary representation and adds that count to itself to obtain the value of the number it supports. That is, if j(m) means the number of ones in the binary representation of m, then m supports m + j(m). Also, numbers that are not supported by any other number are called BLEAK numbers.

Your task is to write a function to recognize if a number is BLEAK or whether it is supported by some other number.

Input Specification:

Input1: An integer value representing the length of input2 array

Input2: An integer array consisting of the integer values

Output Specification:

Return a string array of input1 length which contains the word BLEAK at ith position if input2[i] is bleak or the word SUPPORTED, if input2[i] is supported by another number.

Note: In case of wrong input, the string should contain “-999”

Examples:

2 is not Bleak

3 is not Bleak

4 is Bleak

15 Is Bleak

39 Is Bleak

14 is not Bleak.

8. Bank Problem

There are two banks – Bank A and Bank B. Their interest rates vary. You have received offers from both banks in terms of the annual rate of interest, tenure, and variations of the rate of interest over the entire tenure. You have to choose the offer which costs you least interest and reject the offer. Do the computation and make a wise choice.

The loan repayment happens at a monthly frequency and Equated Monthly Instalment (EMI) is

calculated using the formula given below:

EMI = loan Amount \* monthly Interest Rate / (1 – 1 / (1 + monthly Interest Rate) ^ (number Of Years \* 12))

Constraints:

1<=P<=1000000

1<=T<=50

1<=N1<=30

1<=N2<=30

Input Format:

First line: P principle (Loan Amount)

Second line: T Total Tenure (in years).

Third line: N1 is the number of slabs of interest rates for a given period by Bank A. First slab starts from the first year and the second slab starts from the end of the first slab and so on.

Next N1 line will contain the period and their interest rate respectively.

After N1 lines we will receive N2 viz. the number of slabs offered by the second bank.

Next N2 lines are the number of slabs of interest rates for a given period by Bank B. The first slab starts from the first year and the second slab starts from the end of the first slab and so on.

The period and rate will be delimited by single white space.

Output Format:

Your decision either Bank A or Bank B.

Example 1:

Input:

10000

20

3

5 9.5

10 9.6

5 8.5

3

10 6.9

5 8.5

5 7.9

Output:

Bank B

Example 2:

Input:

500000

26

3

13 9.5

3 6.9

10 5.6

3

14 8.5

6 7.4

6 9.6

Output:

Bank A

9. Implement a stack program using python. Stack functions to be included:

Push

Pop

Get size

10. Implement queue in python.

11. Frequency Stack

Implement a frequency stack with the following methods:

• FrequencyStack() constructs a new instance of a frequency stack

• append(int val) appends val to the stack

• pop() pops and returns the most frequent element in the stack. If there's more than

one most frequent element, the one that's closer to the top of the stack should be

popped first.

You can assume that for pop, the stack is non-empty when they are called.

Constraints

• n ≤ 100,000 where n is the number of methods that will be called to append and pop

Example 1

Input

methods = [""constructor"", ""append"", ""append"", ""append"", ""pop"", ""pop"", ""pop""]

arguments = [[], [1], [1], [2], [], [], []]`

Output

[None, None, None, None, 1, 2, 1]

12. Write a program to sort and merge two linked lists and print the middle element of the merged linked list, if the length is odd, print only the middle element, else if the length is even print the middle 2 numbers.