

1. How to attempt?

Reverse String Word Wise

Write a function to reverse a string word-wise.

Input Specification:

input1: String

Output Specification:

Return the reversed string that is, the last word in input string should come at the first position of the output string, second last word at the second position and so on. Individual words should remain as it is.

Example 1:

Input1: Welcome to mettl

Output: mettl to Welcome

Explanation:

Welcome is the first word of the string and hence is placed at the last position in the output string. Similarly, with to and mettl which get placed at the second and first position respectively in the output string.

Example 2:

Input1: My name is khan

Output: khan is name My

2. Determine how many pairs of students are not following the seating plan.

Write an algorithm to help Rebecca find the number of pairs of students who are not following the seating arrangement.

Input:

The first line of the input consists of two space-separated integers – rows and columns representing the rows (N) and columns (M) of the grid, respectively.

The next N lines consist of M space-separated integers – grid[i][0], grid[i][1],, grid[i][M-1], representing the students sitting in the prescribed arrangement.

Output:

Print an integer representing the number of pairs of persons who are not sitting according to the prescribed arrangement.

Constraints:

$$1 \leq \text{rows} \leq 100$$

$$1 \leq \text{columns} \leq 1000$$

$$-10^9 \leq \text{grid}[i][j] \leq 10^9$$

$$0 \leq i < \text{rows}$$

$$0 \leq j < \text{columns}$$

Note:

If a = b in a pair (a, b), then the pair is considered either increasing or decreasing.

Example:

Input:

2 4
3 4 1 2
1 2 4 3

Output:

9

Explanation:

Step 1: The pairs that are possible in row 1 are:

(3, 4), (3, 1), (3, 2), (4, 1), (4, 2), (1, 2).

The pairs ordered incorrectly are (3, 1), (3, 2), (4, 1), (4, 2).

Step 2: The pairs that are possible in row 2 are:

(1, 2), (1, 4), (1, 3), (2, 4), (2, 3), (4, 3).

The pairs ordered incorrectly are (1, 2), (1, 4), (1, 3), (2, 4), (2, 3).

Step 3: The total number of pairs which are in incorrect order is 4 + 5.

So, the output is 9.

3. Fibonacci

The Fibonacci series is a series in which each number is the sum of the preceding two numbers. For example: 0, 1, 1, 2, 3, 5, 8, 13, 21, etc.

The first two numbers of the series are 0 and 1.

Write a code to return the n^{th} Fibonacci number.

The sequence F_n of Fibonacci numbers is defined by the following recurrence relation $F_n = F_{n-1} + F_{n-2}$ with initial values $F_0 = 0$ and $F_1 = 1$

Input Specification:

Input1: A number n

Output Specification:

Return the n^{th} Fibonacci number

Example 1:

Input1: 2

Output: 1

Explanation:

$$F_n = F_{n-1} + F_{n-2}$$

$$F[2] = [1] + F[0]$$

$$F[2] = 1 + 0$$

$$F[2] = 1$$

Example 2:

Input1: 9

Explanation:

First calculate $F[8]$ and $F[7]$, then

$$F_n = F_{n-1} + F_{n-2}$$

$$F[9] = F[8] + F[7]$$

$$F[2] = 21 + 13$$

$$F[2] = 34$$

4. Longest Increasing Subsequence (LIS)

Given an integer array 'A', find the length of its Longest increasing Subsequence (LIS). LIS is a sub-array of the given integer array where the elements are sorted in a monotonic/strict increasing order.

You need to fill in a function that takes two inputs – integer 'n' and an integer array 'A' containing 'n' integers and returns the length of its LIS.

Input Specification:

Input1: Integer input 'n' ($1 \leq \text{input1} \leq 1000$)

Input2: Integer array 'A' input, containing 'n' integers.

Output Specification:

Return the length of its LIS.

Example 1:

Input1: 3

Input2: {1, 3, 2}

Output: 2

Explanation:

{1, 2} and {1, 3} are the longest increasing subsequences of {1, 3, 2}. The length of both LIS is 2.

Example 2:

Input1: 5

Input2: {41, 18467, 6334, 26500, 19169}

Output: 3

Explanation:

{41, 6334, 26500}, {41, 18467, 26500}, etc are the longest increasing subsequences of {41, 18467, 6334, 26500, 19169}. In all of the cases, the length of LIS is 3.

5. Majority Element

Given 'n' ($1 \leq n \leq 1000$) integers, find the majority element (the integer which occurs more than half the times i.e., occurs more than $n/2$ times). If there is no element that occurs more than $n/2$ times return -1.

Note: The number in the array can contain a negative number also.

Input Specification:

Input1: the number 'n'

Input2: an array of 'n' integers

Output Specification:

Return the majority element or - 1 accordingly.

Example 1:

Input1: 3

Input2: {1, 2, 1}

Output: 1

6. Infix to Prefix conversion

Infix Expressions:

In Infix expressions, operators are written in-between their operands.

An expression such as $A * (B + C) / D$ means –

1. First add B and C together
2. Multiply the result by A
3. Divide by D to get the final answer

The expression for adding the numbers 1 and 2 is “1 + 2”.

Prefix Expressions:

In Prefix expressions, operators are written before their operands.

The expression for adding the numbers 1 and 2 is “+ 1 2”.

In more complex expressions, the operators still precede their operands.

For instance, the expression that would be written in conventional infix notation as

$(5 - 6) * 7$ can be written in prefix as $* - 5 6 7$.

Write a program which takes string as input containing an infix expression and returns the corresponding prefix expression.

Note:

1. The string contains operators (+, -, /, *), parenthesis and operands (**digits**).
2. Each digit is a separate operand.

The operator precedence is as follows:

1. [divide(/) and multiply (*) operator] > [addition(+) and subtraction(-) operator].
2. If you encounter operators with same precedence like (divide and multiply) or (addition and subtraction), then evaluate the operators from left to right.

Input Specification:

Input1: string which contains the infix expression.

Output Specification:

Return the corresponding prefix expression.

Example 1:

Input1: $8 + (7 - 9) * 2$

Output: $+ 8 * - 7 9 2$

Explanation:

$\Rightarrow 8 + (7 - 9) * 2$

=> Reverse is: $2 * (9 - 7) + 8$

1. character: 2	stack: 2	operations: nil
2. character: *	stack: 2	operations:
3. character: (stack: 2	operations: * (
4. character: 9	stack: 2 9	operations: * (
5. character: -	stack: 2 9	operations: * (-
6. character: 7	stack: 2 9 7	operations: * (-
7. character:)	stack: 2 9 7	operations: * (-)
8. character: +	stack: 2 9 7 - *	operations: +
9. character: 8	stack: 2 9 7 - * 8 +	operations: nil

=> Reverse again: $+ 8 * - 7 9 2$

Example 2:

Input1: (1) + 8

Output: +18

Explanation:

=> (1) + 8

=> Reverse is: $8 + (1)$

1. character: 8	stack: 8	operations: nil
2. character: +	stack: 8	operations: +
3. character: (stack: 8	operations: + (
4. character: 1	stack: 8 1	operations: + (

7. Prime Numbers

A prime number is a whole number greater than 1 that is divisible only by itself and the number 1. For example, 2, 3, 5 etc are prime numbers as they are divisible only by themselves and 1.

Given 'n' ($1 \leq n \leq 100000$), find the n^{th} prime number

Input Specification:

Input1: An integer n

Output Specification:

Return the n^{th} prime number

Example 1:

Input: 3

Output: 5

Example 2:

Input: 10

Output: 29

8. Next Greater Number

Given a number 'N' (containing at most 10,000 digits), find the next greater number having the same digits. It is guaranteed that there exists a next greater number having the same digits as N.

Input Specification:

input1: the length of the String 'N'

input2: the number 'N' in the form of a string.

Output Specification:

Return the next greater number having the same digits as 'N' in the form of a string.

Example 1:

input1: 3

input2: 182

Output: 218

Explanation:

Using the same digit the number of permutation are:

1. 128
2. 218
3. 281
4. 812
5. 821

The next greatest number for 182 is 218.

Example 2:

input1: 19

input1: 123456789876554321

Output: 1234567851234456789

Explanation:

The next Greatest number for 1234567849876554321 is 1234567851234456789

Solution

```
import java.util.*;
public class Main
{
    static void swap(char ar[], int i, int j)
    {
        char temp = ar[i];
        ar[i] = ar[j];
        ar[j] = temp;
    }
    static void findNext(char ar[], int n)
    {
        int i;
        for (i = n - 1; i > 0; i--)
        {
            if (ar[i] > ar[i - 1]) {
                break;
            }
        }
    }
}
```



```

    }
    }
    if (i == 0)
    {
        System.out.println("Not possible");
    }
    else
    {
        int x = ar[i - 1], min = i;
        for (int j = i + 1; j < n; j++)
        {
            if (ar[j] > x && ar[j] < ar[min])
            {
                min = j;
            }
        }
        swap(ar, i - 1, min);
        Arrays.sort(ar, i, n);
        System.out.print("Next number with same" + " set of digits is ");
        for (i = 0; i < n; i++)
            System.out.print(ar[i]);
        }
    }
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        String str=Integer.toString(num);
        char digits[] = str.toCharArray();
        int n = digits.length;
        findNext(digits, n);
    }
}
    
```

Python

```
from itertools import permutations
```

```
r=int(input())n=input()
```

```
a = permutations(n, r)
#print(a)

y = [".join(i) for i in a]

y=sorted(y)

q=y.index(n)
#print(q)
print(y[q+1])
```

C++

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;

void swap(char *a, char *b)
{
    char temp = *a;
    *a = *b;
    *b = temp;
}

void findNext(char number[], int n)
{
    int i, j;
    for (i = n-1; i > 0; i--)
        if (number[i] > number[i-1])
            break;
    if (i==0)
    {
        cout << "Next number is not possible";
        return;
    }
    int x = number[i-1], smallest = i;
    for (j = i+1; j < n; j++)
        if (number[j] > x && number[j] < number[smallest])
            smallest = j;
    swap(&number[smallest], &number[i-1]);
    sort(number + i, number + n);
}
```



```
cout << number;

return;
}

int main()
{
    int n;
    cin >> n;
    char digits[n];
    cin >> digits;
    findNext(digits, n);
    return 0;
}
```

9. In a birthday party, the host decides to gift the guest two wins a game in the party. In the game, the host announces an odd number and asks the participants to find all the maximum sized sumfree subsets. The participants will then add all the elements of the sumfree subsets. A set is a sumfree set if no element of the set is the sum of any other two elements in the set. An element can also be considered twice to get the sum. The participant is declared a winner if he/she tells the correct total sum of the elements of all the maximal sized sumfree sets. He/She will get number of gifts equal to the total sum.

Write an algorithm to find the maximum number of gifts a participant can win.

Input:

The input consists of an integer N, representing the odd number announced by the host.

Output:

Print an integer representing the maximum number of gifts a participant can win.

Constraints:

$$1 \leq N \leq 10^6$$

Note:

The output can be large so print the output modulo 1000000007.

10. Maximum Sum

Write a program that adds up the largest row sum and the largest column sum from an N – rows *M-columns array of numbers.

As a preliminary phrase, you should reformat the sequence of numbers as a matrix, whose number of rows and columns are to be specified as arguments.

Input Specification:

Input1: Integer for row dimension of the array

Input2: Integer for column dimension of the array

Input3: Array elements to be entered in row major

Output Specification:

Largest row sum + Largest column sum

Example 1:

Input1 : 2

Input2 : 2

Input3: {1, 2, 5, 6}

Output: 19

Explanation:

The array has 2 rows (input1) and 2 columns (input2). The elements in the first row are 1 and 2 and the elements in the second row are 5 and 6 (input3).

The largest sum among the two rows is 11(5 + 6). The largest sum among the two columns is 8 (2 + 6). By adding those two up, we get the final sum of 19(11 + 8).

Example 2:

Input1 : 3

Input2 : 3

Input3: {3, 6, 9, 1, 4, 7, 2, 8, 9}

Output: 44

Explanation:

In the given matrix of 3×3 , the row with maximum sum has a sum of 19 and the column with the maximum sum has a sum of 25, hence the total sum of the two is: $19 + 25 = 44$

11. Minimum Number of Jumps

Given an array of integers, where each element represents the maximum number of jumps that can be taken forward from that element. Find the minimum number of jumps to reach the end of the array (starting from the first element).

If an element is 0, then no jump can be made from that element. If it is not possible to reach the end, then output -1.

Input Specification:

Input1: number of elements in array input2. ($2 \leq \text{input1} \leq 1000$)

Input2: an integer array

Output Specification:

Return the minimum number of jumps to reach the end.

Example 1:

Input1 : 3

Input2: {2, 1, 1}

Output: 1

Explanation:

The first element is 2, this means that 2 jumps can be taken forward from this element. With this, we reach the last element. Hence, the total number of required jumps is 1.

Example 2:

Input1: 9

Input2: {1, 3, 6, 1, 0, 9, 8, 7, 6}

Output: 3

Explanation:

The first element is 1, this means that 1 jump can be taken and reach element 3. Now, 3 jumps can be taken forward from this element. On taking the first step, we get element 6. If 6 jumps are taken from here itself, the last element of the array will be reached. Hence, the total number of required jumps will be 3.

12. Sum of GP

A geometric progression 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, 1, 3, 9, 27, 81 is a GP with the common ratio as 3.

Given the second and third terms of a GP, return the sum till the n th term of the GP.

The output returned should be a double value representing the sum till the n th term of the GP, rounded off to 3 decimal places.

Input Specification:

Input1: Second Term of GP (Double)

Input2: Third Term of GP (Double)

Input3: Value of n

Output Specification:

Return a double value representing the sum till the n th term, rounded off to 3 decimal places.

Example 1:

Input1: 1

Input2: 2

Input3: 4

Output: 7.5

Explanation:

The sum of the elements of the GP i.e $(0.5 + 1 + 2 + 4)$ 7.5

Example 2:

Input1: 1

Input2: 2

Input3: 5

Output: 15.5

Explanation:

The sum of the elements of the GP i.e $(0.5 + 1 + 2 + 4 + 8)$ 15.5

13. Sum of Divisors

Given an integer ' n ' ($1 \leq n \leq 10^9$), find the sum of its unique divisors.

Input Specification:

Input 1: the integer ' n '

Output Specification:

Return the sum of divisors of ' n '.

Example 1:

Input 1: 6

Output: 12

Explanation:

Divisors of 6 are 1, 2, 3 and 6. Sum of number (i.e 1 + 2 + 3 + 6) is 12

Example 2:

Input1: 36

Output: 91

14. Given three numbers b, e, and m. Fill in a function that takes these three positive integer values and outputs b^e mod m.

Note: The test case values can be very large. Therefore, write your code accordingly.

Input Specification:

Input 1: positive integer, b

Input 2: positive integer, e

Input 3: positive integer, m

Output Specification:

Return an integer on calculating b^e mod m.

Example 1:

Input 1: 2

Input 2: 10

Input 3: 1025

Output: 1024

15. Write a program to return the difference between the sum of odd numbers and even numbers from an array of positive integers.

Note: You are expected to write code in the findOddEvenDifference function only which will receive the first parameter as the number of items in the array and second parameter as array itself. You are not required to take input from the console.

Example

Finding the difference between the sum of odd and even numbers from a list of 5 numbers

Input

Input 1: 5

Input 2: 10 11 7 12 14

Output

-18

Code:

Using System;

Using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int findOddEvenDifference(int input1, int[] input2)

{

//Read only region end

//Write code here

```
        Throw new NotImplementedException("Method findOddEvenDifference (int input1,))
    }
}
```

16. How to attempt?

Placement Season Begins

The placement season has begun in a college. There are N number of students standing outside an interview room in a line. It is given that a person who goes in first has higher chances of getting selected.

Each student has a number associated with them known as the problem-solving capability (PSC). The higher the capability, the higher the chances of selection. Now, each student wants to know the number of students ahead of him/her who have more problem-solving capability than him/her.

Find this number for each student.

Input Specification:

input1: An integer N, which denotes the number of students present.

input2: An array of size N, denoting the problem-solving capability of the students

Output Specification:

An array of size N denoting the required answer for each student.

Example 1:

input1: 6

input2: {4, 9, 5, 3, 2, 10}

Output: {0, 0, 1, 3, 4, 0}

17. You are required to implement the following function:

```
int MatchString(char str1[], int len1, char str2[], int len2, int k1, int k2);
```

The function accepts two strings 'str1' and 'str2' of length 'len1' and 'len2' respectively and two integers 'k1' and 'k2' as its arguments. Implement the function to compare each index of 'str1' and 'str2' leaving out the first 'k1' characters from 'str1' and 'k2' characters from 'str2' respectively till the end of each string and return an integer as per the following rules in the given priority:

1. If all the remaining characters match, then return the length of the match
2. Return 0, if remaining characters to be matched for either of the string is 0 or both the strings are null (or None in case of python)
3. Return -1, in case of mismatch in characters or count of remaining characters or one of the string, is null (or None in case of python)

Assumption:

- 'str1' and 'str2' contain lower case alphabets only
- Index starts from 0
- $0 \leq k1 \leq len1$
- $0 \leq k2 \leq len2$

Example:

Input:

k1: 4

k2: 7

str1: succeed

str2: crossbreed

Output:

3

Explanation:

Leaving first the 4 characters in 'succeed', string left is 'eed', similarly after leaving first the 7 characters in 'crossbreed', string left is 'eed'

Since all the remaining characters match and the length of the match is 3, hence the output is 3.

Sample Input:

k1: 5

k2: 5

str1: remember

str2: customer

Sample Output

- 1

```
int MatchString(char str1[], int len1, char str2[], int len2, int k1, int k2);
```

```
int main()
```

```
{
```

```
//Input read from STDIN
```

```
int result = MatchString(str1, len1, str2, len2, k1, k2);
```

```
//Value in result printed to STDOUT
```

```
return 0;
```

```
}
```

18. How to attempt?

Infix to Postfix Conversion

Infix Expressions:

In Infix expressions, operations are written in-between their operands.

An expression such as $A * (B + C) / D$ means-

1. First add B and C together

2. Multiply the result by A

3. Divide by D to get the final answer

The expression for adding the numbers 1 and 2 is " $1 + 2$ ".

Postfix Expressions:

In Postfix expressions, the operators follow their operands.

The expression for adding the numbers 3 and 4 is " $3 4 +$ ".

If there are multiple operations, the operator is given immediately after its second operand.

So the expression written for " $3 - 4 + 5$ " would be " $3 4 - 5 +$ " in Postfix notation. Here, 4 is first subtracted from 3, then added to 5.

Write a program that takes input as a string containing an infix expression and returns the corresponding postfix expression.

Note:

1. The string contains operators (+, -, /, *), parenthesis and operands. (digits).

2. Each digit is a separate operand.

The operator precedence is as follows:

19. How to attempt?

Primes with a Twist

Given an integer n ($n \leq 10^4$), you need to count the numbers, $x_i < n$, which are co-prime to 'n', i.e. $\gcd(x, n) = 1$.

Formally, given n , you need to find $f(n) = |\{x < n : \gcd(x, n) = 1\}|$.

Input Specification:

input1: the integer 'n'

Output Specification:

Return the count of the number of co-primes of 'n'.

Example 1:

input1: 4

Output: 2

Explanation:

Integers 1 and 3 are co-prime to 4, but 2 is not.

Example 2:

input1: 16

Output: 8

Explanation:

Integers 1, 3, 5, 7, 9, 11, 13 and 15 are co-prime to 16.

20. How to attempt?

String Permutations

You are given two strings 'X' and 'Y', each containing same no of characters.

Write a program that can determine whether the characters of string 'X' can be rearranged to form the second string 'Y'. print "yes" if this is possible and "no" if not.

Input Specification:

input1: the string 'X'

input2: the string 'Y'

Output Specification:

Return "yes" or "no" accordingly.

Example 1:

input:

z b k

z k b

Output: yes

Explanation:

You can rearrange zbk to be zkb (by switching the characters, output is "Yes".)

Example 2:

Input:

sample

pleamc

Output: no

Explanation:

You can not rearrange "pleam" to be "sample" (output is "No".)

21. Java's BigDecimal class can handle arbitrary-precision signed decimal numbers. Let's test your knowledge of them!

Given an array n of real number strings, sort them in descending order.

Complete the code in the unlocked section of the editor below. You must rearrange array's elements according to the given instructions.

Input Format:

The first line consists of a single integer, n, denoting the number of integer strings.

Each line i of the n subsequent lines contains a real number denoting the value of s_i.

Constraints:

1 <= n <= 200

Each s_i has at most 300 digits.

Output Format:

Locked stub code in the editor will print the contents of array to stdout. You are only responsible for reordering the array's elements in descending order.

Sample Input:

```
10
9
- 100
50
0
56.6
90
0.12
0.16
02.34
000.000
```

Sample Output:

```
90
56.6
50
9
2.34
0.16
0.12
0
0.000
- 100
```

22. How to attempt?

Remainder mod 11

Given a string (of maximum length 1000) representing a large number, output its remainder modulo 11.

Input Specification:

Input1: a large number in the form of a string.

Output Specification:

Return the remainder modulo 11 of input1.

Example 1:

input1: 121

Output: 0

Explanation:

$121 \bmod 11 = 0$

Example 2:

input1: 452

Output: 1

Explanation:

$452 \bmod 11 = 1$

23. How to attempt?

The Cuckoo Sequence

A Cuckoo Sequence is defined as shown.

$\text{Cuckoo}[1] = 0$

$\text{Cuckoo}[2] = 1$

$\text{Cuckoo}[n] = 1 * \text{Cuckoo}[n - 1] + 2 * \text{Cuckoo}[n - 2] + 3 * 1$, for $n > 2$

Given n ($1 \leq n \leq 10^9$), find $\text{Cuckoo}[n]$.

Input Specification:

input1: Integer 'n'

Output Specification:

Return the value of $\text{Cuckoo}[n]$.

Example 1:

input1: 3

Output: 4

Explanation:

$\text{Cuckoo}[n] = 1 * \text{Cuckoo}[n - 1] + 2 * \text{Cuckoo}[n - 2] + 3 * 1$

$\text{Cuckoo}[3] = 1 * \text{Cuckoo}[2] + 2 * \text{Cuckoo}[1] + 3 * 1$

$\text{Cuckoo}[3] = 1 * 1 + 2 * 0 + 3 * 1$

$\text{Cuckoo}[3] = 4$

Example 2:

input1: 2

Output: 1

Explanation:

$\text{Cuckoo}[2] = 1$ as given in the question.

24. How to attempt?

Arithmetic Progression

Given the second and the third terms of an AP ($-10^6 \leq a^2, a^3 \leq 10^6$), find the n^{th} ($1 \leq n \leq 1000$) term of the sequence.

Input Specification:

input1: Second element of series (Integer).

input2: Third element of series (Integer).

input3: Total number of elements in the series (Integer).

Output Specification:

Return the n^{th} element of the series.

Example 1:

input1: 1

input2: 2

input3: 4

Output: 3

Explanation:

$a_2 = 1, a_3 = 2, n = 4, d = 1, a_n = a_4 = 3$ (d refers to the common difference between adjacent terms in an arithmetic progression)

Example 2:

input1: 5

input2: 8

input3: 4

Output: 11

Explanation:

$a_2 = 5, a_3 = 8, n = 4, d = 3, a_n = a_4 = 11$ (d refers to the common difference between adjacent terms in an arithmetic progression)

25. How to attempt?

Anagrams

An anagram is a word, phrase, or name formed by rearranging the letters of phrase, or name.

Write a function to check if two given strings are anagrams or not. Return "Yes" anagrams, otherwise return "no".

Input Specification:

input1: the first string

input2: the second string

Output Specification:

Return "yes" if they are anagrams, otherwise return "no".

Example 1:

input1: build

input2: dubli

Output: yes

Explanation:

First string can be rearranged to form the second string other.

Example 2:

input1: beast

input2: yeast

Output: no

26. How to attempt?

Numbers Puzzle

Given a set of numbers, one can arrange them in any order but must pay a penalty equal to the sum of the absolute differences between adjacent numbers.

Return the minimum penalty that must be paid.

Input Specification:

input1: length of an integer array of numbers ($2 \leq \text{input1} \leq 1000$)

input2: integer array ($1 \leq \text{input2}[i] \leq 10000$)

Output Specification:

Return the minimum penalty.

Example 1:

input1: 3

input2: {1, 3, 2}

27. Write a program that takes in input as String x and returns the length of the longest palindrome subsequence of x.

Input Specification:

input1: string input

Output Specification:

Return the length of the longest palindromic subsequence.

Example 1:

input1: ababa

Output: 5

Explanation:

Length of Longest palindromic sequence is 5 that is "ababa"

Example 2:

input1: umeaylnlfd

Output: 3

Explanation:

Length of Longest palindromic sequence is 3 that is "lnl".

28. Create a program that take a name as input and returns a greeting.

Examples Input:

Gerald

Example Output:

Hello Gerald!

Examples Input:

Tiffany

Example Output:

Hello Tiffany!

29. How to attempt?

Highest Common Factor

Find the HCF (Highest Common Factor) of n numbers given in an integer array. Fill in the function HCF() and return the HCF.

Input Specification:

input1: the size array

input2: an integer array

Output Specification:

Return the HCF of given numbers

Example 1:

input 1: 3

input 2: {2, 4, 8}

Output: 2

Explanation:

The common factor for 2, 4, 8 are 1 and 2. Hence the HCF(Highest Common Factor) is 2.

Example 2:

input1: 5

input2: {10, 15, 20, 35, 70}

Output: 5

30. OTT Cracker

You are a OTT cracker, who cracks and sell OTT accounts for cheap, through software called OTT checker. To avoid getting caught, gives a password for each email in the form of a code and you have to crack it. Now, the code is divided into parts and is represented is an array, and the number of parts of code is the size of the array. Now, you need to convert each part of code into a number and then you need to combine the parts of the array which are now numbers in such a way that the final number obtained after merging all elements of the array is the maximum possible number that can be formed from them.

For code conversion, you need to follow excel sheets, say array element is **A**, it will be numbered 1, if the element is **Z**, it will be numbered **26**, **AA** will be numbered **27**, and so on. Now, after code conversion, say you get **3** elements, these have to be combined to form the maximum possible number.

Example

Assumptions

- N = 2
- Array of string's = {"AZ", "ZA"}

Approach

There are 2 parts of code:

"AZ" maps to 52 and "ZA" maps to 677

So, maximum number possible after combining 52 and 677 comes out to be 67752 which is the cracked password here.

Function description

Complete the solve function provided in the editor. This function takes the following 2 parameters and returns the final password.

- N: Represents the number of parts in which code is divided

- code_part: Represents the array of strings each containing only upper-case English letters, denoting N part of code.

Input Format-

Note: This is the input format that you must use to provide custom input (available above the **Compile and Test** button).

The first line contains an integer, N denoting the number of parts in which code is divided.

The next line contains N space-separated string each containing only upper-case English letters, denoting N part of code.

Output Format-

The final password should be printed.

Sample input 1:

3

AZ BA P

Sample output 1:

535216

Explanation:

- N = 3
- Array of string's = ["AZ", "BA", "P"]

Approach

There are 3 parts of code:

AZ, BA and P

After conversion,

AZ = 52, BA = 53 and P = 16

So, maximum number possible after combining 52, 53 and 16 comes out to be 535216 which is the cracked password here.

Note: Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore, your code must pass these hidden test cases to solve the problem statement.

Time Limit: 5.0 sec(s) for each input file

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Score is assigned if any test case passes

- An **Autobiographical number** is a number N such that the first digit of N represents the count of how many zeroes are there in N, the second digit represents the count of how many ones are there in N and so on.

You are given a function,

in FindAutoCount(char* n);

The function accepts string "n" which is a number and checks whether the number is an autobiographical number or not. If it is an autobiographical number it returns an integer, i.e. the count of distinct numbers in "n". If it is not an autobiographical number it returns 0.

Assumption:

- The input string will not be longer than 10 characters.
- Input string will consist of numeric characters

Note:

If string is null return 0.

Example:

Input:

n : "1210"

Output:

3

Explanation:

0th position in the input contains the number of 0 present in input i.e. 1, in 1st position the count of number of 1's in input i.e. 2, in 2nd position the count of 2's in input i.e. 1, and in 3rd position the count of 3's i.e. 0, so the number is autobiographical number.

Now unique numbers in the input are 0, 1, 2, so count of unique numbers are 3. So 3 is returned.

Sample Input:

n : "123"

Sample Output:

0

32. You are required to implement the following function:

int CountKDigitNumbers(int arr[], int n, int k);

The function accepts an array 'arr' of 'n' integers and an integer 'k' as its arguments. You are required to calculate the number of 'k' digit integers in array 'arr', and return the same.

Note:

- n > 0
- k > 0

Example:

Input:

arr: 1 2 22 3 34 899 112 3 44 552

n: 10

k: 2

Output:

3

Explanation:

2 digit integers in array 'arr' are {22, 34, 44}, count of them is 3, hence 3 is returned.

Sample Input

arr: 332 1 302 41 44 95 122 85 65 3221 775 12

n: 12

k: 3

Sample Output:

4

33. Implement the following function:

int MaximaOrMinima(int a, int b, int c);

Quadratic equation: A quadratic equation is any equation having the form, $ax^2 + bx + c$, where 'a' cannot be zero.

The function accepts coefficients of a quadratic equation, 'a', 'b' and 'c' as its argument. Implement the function to find the maximum or minimum value of quadratic equation by substituting integer values x, where $-100 \leq x \leq 100$. Return the values as follows:

- if $a > 0$, return the minimum value of the equation.
- if $a < 0$, return the maximum value of the equation.

Assumption: a is not equal to zero.

Note: Computer value lies within integer range

Example:

Input:

a: 1

b: 2

c: 1

Output:

0

Explanation:

Since, ($a > 0$) output is the minimum value which is at $x = -1$, output = $1 * (-1)^2 + 2 * (-1) + 1 = 0$.

Sample Input:

a: -2

b: -8

c: 10

Sample Output:

18

34. Implement the following function:

int MaxProfit(int* price, int n);

The function accepts a positive integer array 'price' consisting of 'n' elements as its argument. 'price' contains prices of a share throughout the day in the order of their occurrence. When a new price occurs, old price disappears forever. A transaction for the share is complete when firstly the share is bought and the sold. In a transaction where the share is bought at price[j] and sold at price[k], where $j < k$, profit for that transaction = price[k] - price[j]. You can use a price only once either to buy or sell the share. Implement the function to maximize the profit in at most 2 transactions and return the same. Transaction 2 starts only when transaction 1 is completed.

Note:

- If 'price' is empty or None in case of Python, return -1.
- Return 0, if $n < 2$.
- All calculations lie within integer range.

Example:

Input:

price: 2 30 15 10 8 25 80

Output:

100

Explanation:

- For Transaction 1, share is bought at 2 and sold at 30. Profit = $30 - 2 = 28$.
- For Transaction 2, share is bought at 8 and sold at 80. Profit = $80 - 8 = 72$.
- Total profit = Transaction 1 profit + Transaction 2 profit = $28 + 72 = 100$.

Thus, output is 100.

Sample Input:

price: 10 5 22 65 8 75 90 80

Sample Output:

142

35. How to attempt?

Documents

The United Nations Organization released an official document regarding the most important events from the beginning of time (dated 00-00-0000) with a brief description of the events. The date of all the events is mentioned in the 'DD-MM-YYYY' format.

Find the total number of distinct years referenced in the document.

Input Specification:

input1: String containing the content of the document

Output Specification:

Return the total number of distinct years referenced in the document.

36. There is a singly linked list represented by the following structure:

```
struct Node
{
    int data;
    struct Node* next;
};
```

Implement the following function:

```
struct Node* DeleteNodes(struct Node* head);
```

The function accepts a pointer to the start of the linked list, 'head' as its argument. Delete all such nodes from the training list whose adjacent node on the right side has greater value and return the modified linked list.

Note:

- Return null if the list is empty (Incase of python if the list is None return None).
- Do not create a new linked list, just modify the input linked list and return it.

Example:

Input:

head: 6 -> 2 -> 5 -> 4 -> 9 -> 7 -> 2 -> 1 -> 5 -> 9

Output:

6 -> 5 -> 9 -> 7 -> 2 -> 9

Explanation:

Node '2' is deleted as '2' < '5' then '4' is deleted as '4' < '9' then '1' is deleted as '1' < '5' then '5' is deleted as '5' < '9'.

Sample Input:

head: 9 -> 5 -> 6 -> 2 -> 7

Sample Output:

9 -> 6 -> 7

37. Demolition Robot

You are in charge of preparing a recently purchased lot for one of Amazon's new building. The lot is covered with trenches and has a single obstacle that needs to be taken down before the foundation can be prepared for the building. The demolition robot must remove the obstacle before progress can be made on the building.

Write an algorithm to determine the minimum distance required for the demolition robot to remove the obstacle.

Assumptions:

The lot is flat, except for trenches, and can be represented as a two-dimension grid.

The demolition robot must start from the top-left corner of the lot, which is always flat, and can move one block up, down, left, or right at a time.

The demolition robot cannot enter trenches and cannot leave the lot.

The flat areas are represented as 1, areas with trenches are represented by 0 and the obstacle is represented by 9.

Input:

The input to the function/method consists of one argument:

lot, representing the two-dimensional grid of integers.

Output:

Return an integer representing the minimum distance traversed to remove the obstacle else return -1.

Constraints

$1 \leq \text{rows, columns} \leq 10^3$

Example:

Input:

lot =

[[1, 0, 0],

[1, 0, 0],

[1, 9, 1]]

Output:

3

Explanation:

Starting from the top-left corner, the demolition robot traversed the cells (0, 0) -> (1, 0) -> (2, 0) -> (2, 1).

The robot traversed the total distance 3 to remove the obstacle.

So, the output is 3.

38. Single File Programming Question

Write a Python program that accepts a string and calculate the number of upper case letters and lower case letters.

Input Format:

A string in the first line

Output Format:

Print the original string in the first line.

Number of upper case characters in the second line

Number of lower case characters in the third line

Refer to the sample output for the exact format

Sample testcases

Input 1:

The quick Brown Fox

Output 1:

The quick Brown Fox

Upper case characters : 3

Lower case characters : 13

39. Array Operations: At least M Balls

You are given N Bags each containing some balls. You need to find how many extra balls you will need to maintain at least M balls in each bag.

Note:

Balls already in the bag cannot be taken back

Function Description

In the provided code snippet, implement the provided findBalls(...) method using the variables to print the output. You can write your code in the space below the phrase "WRITE YOUR LOGIC HERE".

There will be multiple test cases running so the Input and Output should match exactly as provided.

The base Output variable result is set to a default value of - 404 which can be modified. Additionally, you can add or remove these output variables.

Input Format:

The first line contains 2 space-separated integer N and M denoting the number of bags and the minimum number of balls to be placed respectively.

The second line contains array A denoting the number of balls in each bag.

40. The current selected programming language is C. We emphasize the submission of a fully working code over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use printf() to debug your code. The printf() may not work in case of syntax/runtime error. The version of GCC being used is 5.5.0.

The city authorities are examining the houses in a residential area for a city planning scheme. The area is depicted in an aerial view and divided into a $N \times M$ grid. If a grid cell contains some part of a house roof, it is assigned a value 1; if not, then the cell represents a vacant plot and is assigned the value 0. Clusters of adjacent grid cells with value 1 represent a single house. Diagonally placed grids with value 1 do not represent a single house. "Beautiful house" is a special kind of house that is surrounded by vacant plots on all sides: horizontally, vertically and diagonally. You may assume that all four boundaries of the given grid are surrounded by vacant plots.

Write an algorithm to determine the number of "beautiful houses".

Input:

The first line of the input consists of two space-separated integers – rows and cols representing the number of rows (N) and the number of columns in the grid (M), respectively.

The next N lines consist of M space-separated integers representing the grid.

Output:

Print an integer representing the number of "beautiful houses".

Constraints:

The elements of the grid consist of 0s and 1s only.

Examples:

Example 1:

Input:

```
4 4
1 0 1 1
0 0 1 0
1 0 1 1
1 0 1 0
```

Output:

3

Explanation:

There are 3 beautiful houses that are completely surrounded by vacant plots.

So, the output is 3.

Example 2:

Input:

```
5 7
1 0 0 0 0 1 1
0 1 0 0 0 1 0
0 0 1 0 0 1 1
1 0 0 1 0 1 1
1 0 0 0 0 0 0
```

Output:

2

Explanation:

There are 2 beautiful houses that are completely surrounded by vacant plots.

So, the output is 2.

41. Valid Email Addresses

Email addresses are of the form [user@domain.extension](#)

Valid HackerRank emails are of the form [user@hackerrank.com](#), and the characteristics of user are:

- It starts with 1 to 6 lowercase English letters denoted by the character class [a-z].
- The lowercase letter(s) are followed by an optional underscore, i.e. zero or one occurrence of the underscore '_' character.
- The optional underscore is followed by 0 to 4 optional digits denoted by the character class [0-9].

Complete the code in the editor below by replacing the blank ("_____") with a regular expression that matches a valid email addresses according to the criteria above. Locked code in the editor prints True for each correct match and False for each incorrect match.

An example of a valid email is [abcdef 1234@hackerrank.com](#). It has as many of each class of character as possible. The address [a1_1@baddomain.com](#) fails for two reasons. First, digits cannot precede the underscore. Second, the domain fails because it is not hackerrank.com

Constraints:

- $1 \leq \text{query} \leq 10^3$
- $1 \leq \text{string length} \leq 10^3$.

Input Format:

Sample Case 0:

STDIN

Function

-

5

→ query =

5

[julia @hackerrank.com](#) → each of these strings is queried via regex

[julia_0@hackerrank.com](#)

[julia0 @hackerrank.com](#)

[julia@gmail.com](#)

Sample Output: 0

True

True

True

False

False

Explanation: 0

The following query = 5 validations are performed:

1. [julia@hackerrank.com](#) starts with between 1 and 6 lowercase letters and contains zero of the optional characters, so it is valid.
2. [julia @hackerrank.com](#) starts with between 1 and 6 lowercase letters, is followed by a single underscore, and contains none of the optional digits, so it is valid.
3. [julia_0@hackerrank.com](#) starts with between 1 and 6 lower case letters, is followed by a single underscore, and is followed by between 0 and 4 digits, so it is valid.
4. [julia0 @hackerrank.com](#) has valid lowercase letters followed by a valid digit, but the digit must not precede the underscore.
5. [julia@gmail.com](#) has valid user name but its domain and extension do not match hackerrank.com.

42. Implement the following function:

```
static int CountPoints(int[] circle1, int[] circle2){}
```

The function accepts two arrays 'circle1' and 'circle2' of size 3. Each array is defined as {a, b, r}, where (a, b) is the center point of the circle and r is its radius, these circles lie on the positive xy-plane. Implement the function to find and return the number of integral points that lie on or inside the common region of two circles.

$$\text{Circle: } (a - x_1)^2 + (b - y_1)^2 = r^2$$

Assumption: circle1[0] = a, circle1[1] = b and circle1[2] = r, similarly for circle 2.

Note:

- Return 0 if no common point is found.
- Arrays are not null.

Example:

Input:

a b r

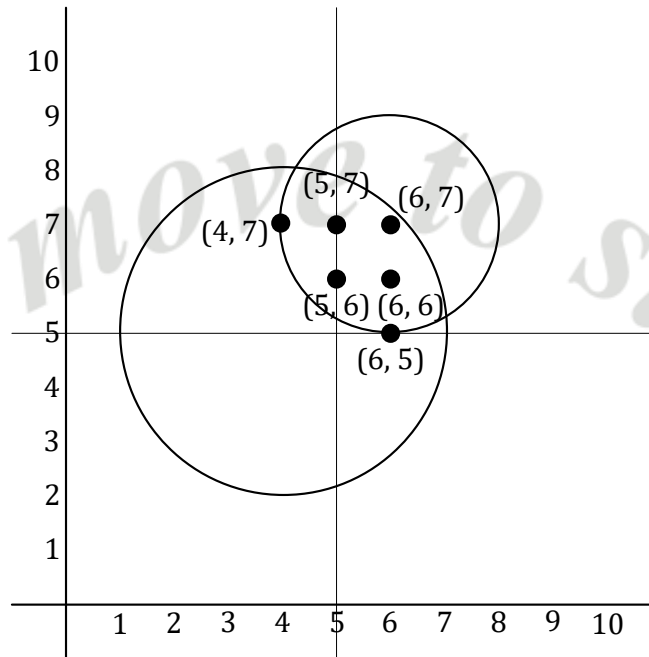
circle1: 4 5 3

circle2: 6 7 2

Output:

6

Explanation:



6 points that lie on or inside the common region of two circles are (4, 7), (5, 7), (6, 7), (5, 6), (6, 6), and (6, 5)

Sample Input:

a b r

circle1: 3 3 2

circle2: 5 3 2

Sample Output:

5

43. Number of Houses

Problem Statement

Implement the following function:

```
int NumberofHouses(int r, int unit, int arr[], int n);
```

The function accepts two positive integers 'r' and 'unit' and a positive integer array 'arr' of size n as its argument. 'r' represents the number of rats present in an area, 'unit' is the amount of food each rat consumes and each i^{th} element of array 'arr' represents the amount of food present in 'i + 1' house number, where $0 \leq i < n$. Implement the function to find and return the house number, starting from 1st house (with the amount of food arr[0]), till which the amount of food is sufficient for all the rats.

Note:

- Return - 1 if the array is null (or None in the case of python).
- Return 0 if the total amount of food from all houses is not sufficient for all the rates.

- Computed values lie within the integer range.

Example:

Input:

r: 7

unit: 2

arr: 2 8 3 5 7 4 1 2

Explanation:

Total amount of food required for all rats = $r * \text{unit} = 7 * 2 = 14$. Total amount of food in 1st four houses = $2 + 8 + 3 + 5 = 18$. Since, amount of food in 1st four houses is sufficient for all rats. Thus, output is 4.

Sample Input:

r: 4

unit: 3

arr: 5 6 1 5 7

Sample Output:

3.

44. Palindrome Count

Write a function to find all the words in a string which are palindrome.

Note: A string is said to be a palindrome if the reverse of the string is the same as string. For example, "abba" is a palindrome, but "abbc" is not a palindrome.

Input Specification:

Input1: string

Input2: Length of the String

Output Specification:

Return the number of palindromes in the given string

Example 1:

Input1: this is level 71

Input2: 16

Output: 1

Explanation:

The reverse of the word "level" is "level". Hence, the word is a palindrome. As the string contains only one palindrome, so the returned value will be 1.

Example 2:

Input1: hello world

Input2: 11

Output: 0

Explanation:

As the given string doesn't contain any palindrome, so the returned value will be 0.

45. Input Format:

The first line contains three integers n , x_0 and y_0 ($1 \leq n \leq 1000$, $-10^4 \leq x_0, y_0 \leq 10^4$) — the number of storm troopers on the battle field and the coordinates of your gun.

Next n lines contain two integers ($-10^4 \leq x_i, y_i \leq 10^4$) each represent— he coordinates of the storm troopers on the battlefield. It is guaranteed that no storm trooper stands at the same point with the gun. Multiple storm troopers can stand at the same point.

Output Format:

Print a single integer — the minimum number of shots Han Solo needs to destroy all the stormtroopers.

Examples:

Input:

4 0 0

1 1

2 2

2 0

-1 -1

Output:

2

Input:

2 1 2

1 1

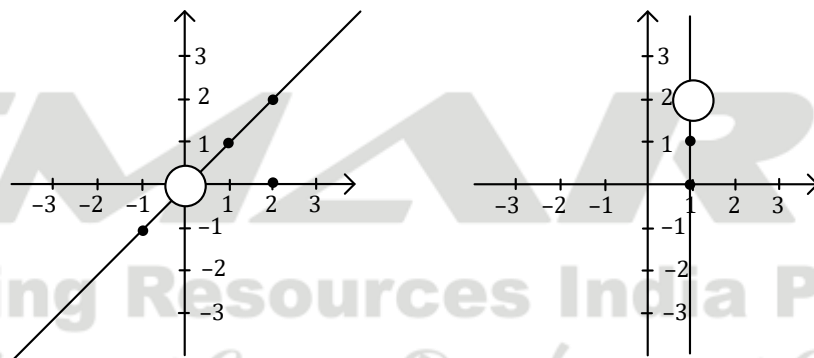
1 0

Output:

1

Note:

Explanation to the first and second samples from the statement, respectively:



46. Write a program to take two integers m & n as input and find the number of possible sequences of length n such that each of the next elements is greater than or equal to twice of the previous element but less than or equal to m .

Example 1:

Input:

10

4

Output: 4

Explanation: There should be n elements and value of last element should be at-most m .

The sequences are $\{1, 2, 4, 8\}$ $\{1, 2, 4, 9\}$, $\{1, 2, 4, 10\}$, $\{1, 2, 5, 10\}$.

Example 2:

Input:

5

2

Output: 6

Explanation: The sequences are {1, 2}, {1, 3}, {1, 4}, {1, 5}, {2, 4}, {2, 5}.

Expected Time Complexity: $O(m*n)$

Expected Auxiliary Space: $O(1)$

Constraints: $1 \leq m, n \leq 100$

47. Write a Program TO find SUM of ALL integers BETWEEN two integer numbers taken as input AND are divisible BY 7.

Constraint:

Input1 < Input2

Example Input:

1

20

Example Output:

21

48. Please do not forget to select any one language from the right, under "Select Language" drop down menu before you start coding. This is a programming question. Please write your program in the given text area. You can check your program by clicking on the "Compile" button. Once your program is finalized, you must submit your program for evaluation by clicking on the "Submit" button. Also ensure that while printing the output of your program, if required, you only print the exact output of the program, without any leading or preceding text or remark.

Dice sum:

Here you are given 2 unbiased dice containing 6 faces. You will be given an output sum which should be obtained by throwing two dice. You need to return the number of all possibilities where the sum on both the dice is equal to the output sum. If there are no possibilities return 0.

Sample Input:

10

Output:

3

Explanation:

The possible outcomes with the output sum of 10 is (5, 5), (6, 4) and (4, 6)

49. Please do not forget to select any one language from the right, under "Select Language" drop down menu before you start coding. This is a programming question. Please write your program in the given text area. You can check your program by clicking on the "Compile" button. Once your program is finalized, you must submit your program for evaluation by clicking on the "Submit" button. Also ensure that while printing the output of your program, if required, you only print the exact output of the program, without any leading or preceding text or remark.

Word play-2:

You will be given a $n*n$ matrix with jumbled characters. You need to find and return the word "Welcome" such that the word formed from the letters in the matrix are sequentially adjacent, adjacent cells means those which are horizontally or vertically neighboring. If the word is not formed return False.

Input Format:

First line has "n" which indicates the rows and columns of the grid.

Second line has all the matrix elements divided with one space.

Output Format:

String 'welcome' if the word is forms otherwise False.

Sample Input:

4

a b c o d e l m q w q e z x y k

Output: welcome

Explanation: The matrix formed is [[a, b, c, o].

[d, e, l, m]

[q, w, q, e]

[x, y, z, K]].

The word can be formed form:

w(2, 1) -> e(1, 1) -> l(1, 2) -> c(0, 2) -> o(0, 3) -> m(1, 3) -> e(2, 3).

50. Binary Tree Combinations:

Given the elements of binary tree in an array format. You need to return the number of possible ways to reorder the elements in the array such that the binary tree is similar to the older one.

Sample Input: 4 3 5

Output: 1

Explanation:

4

/ \

3 5

[4, 5, 3] is the only possible way to reorder such that the binary tree doesn't change.

Sample Input: 1 2 3

Output: 0

Explanation: 1

2

3

No possible way to reorder

51. Elevator Installation:

Ravi wants to install an elevator at his home for domestic uses. He has everything except equal weight boxes to balance it on the either side(two weights) of the elevator. Help Ravi installing the elevator with the multiple weight boxes he has by returning the final weight which can be formed by merging possible unequal weights. The only condition here is the final weight boxes should be of equal weights. Return the maximum possible final weight so maximum possible final weight so that elevator can be more balanced.

Sample Input:

1 2 3

Output:

3

Explanation: Here to balance the weights the only possible way is to merge the weights 1 and 2, such that we can form two final weight boxes of 3.

52. Given a set of points in a plane, check whether the points lie on a straight line or not. If they lie on a straight line return the equation else return 0.

Input format:

First line contains no. of points in the plane.

Second line contains x, y coordinates of all the points which are divided with spaces.

Output format:

Equation of the line(str) if the points lie on a plane or 0(str).

Sample Input:

3

1 1 2 2 3 3

Output: $1x - 1y = 0$

Explanation: The three points here are [1, 1], [2, 2] and [3, 3]. These lie on a line so it returned the equation.

53. Step game:

This is a game where there are steps numbered from 0 to the n steps. You will be on the nth step on the start of the game. Your goal is to reach the 0th step at the end of the game with minimum no. of jumps. If the given step's number is even you are allowed to jump $m/2$ steps below at maximum and if the step no. is odd you are allowed to jump 1 step below. Now find the minimum number of steps required to win this game from the given input.

Sample Input:

10

Output:

5

Explanation:

10 -> 5 jump - 1

5 -> 4 jump - 2

4 -> 2 jump - 3

2 -> 1 jump - 4

1 -> 0 jump - 5

Write an algorithm to solve it.

54. Write a program to input string and display count of all permutations of string without using any built-in functions.

Example Input:

ABC

Example Output:

6

Explanation:

Total Permutation of the string can be: ABC ACB BAC BCA CAB CBA

and their count is 6 which is why answer is 6

55. Write a program to take string as input and check if it contains only characters or not, numeric and special symbols are not allowed. After that, reverse that string without using built-in functions. If entered and reverse

string both are same than give message as provided in double quotes "you inputted a strong string", otherwise print "you inputted it weak string."

Example Input:

wowu

Example Output:

you inputted a weak string

Example Input:

wow

Example Output:

you inputted a strong string.

Smart move to success

SMART

Training Resources India Pvt Ltd
India's Largest Career Development Company