Accenture

You are given a number N. Convert the number to its binary form and return the sum of its binary digits (i.e., the number of 1's in the binary representation)

Input Format:

Input Contains an integer digit n

Output Format:

It represents the sum of the number's binary digits

Input:
15
Output:
4

Explanation:
Binary of 15 is 1111, so the sum is 4.

Given a Word, return a string that contains the first letter, the count of the middle characters, and the last letter.

Input Format:

Input Contains a String s.

Output Format:

Refer the Sample Output.

Input:
examination

Output:
e9n

Given a **list of numbers**, the task is to find and return a list with the cumulative sum of given list.

As per the problem statement we have to find a cumulative sum array which means array elements will be updated with the sum of the current element and all previous elements.

Input Format:

The First line represents the integer size n.

The next line represents n space separated integer values

Output Format:

It prints the updated elements in an array.

Input:
7
2 9 11 5 15 6 10
Output:
2 11 22 27 42 48 58

Given string **str** of alphanumeric characters, the task is to check whether the string is a valid **PAN (Permanent Account Number) Card** number or not by using Regular Expression.

The valid PAN Card number must satisfy the following conditions:

It should be ten characters long.

The first five characters should be any upper case alphabets.

The next four-characters should be any number from 0 to 9.

The last(tenth) character should be any upper case alphabet.

It should not contain any white spaces.

Input 1:
BNZAA2318J
Output 1:
Valid PAN no
Input 2:
23ZAABN18J
Output 2:
Invalid PAN no

Given an array **arr[]** of **N** integers. The task is to find the Sum of factorial of positive and single digit numbers in an array.

Input Format:

First line represents the array size n.

The next n line represents the elements in an array.

Output Format:

It represents the sum of factorial of positive and single digit number.

Input 1:
5
7 3 5 4 8
Output 1:
45510

In an array a superior element is one which is greater than all elements to its right. The rightmost element will always be considered as a superior element.

You are given a function,

int FindNumberOfSuperiorElements(int* arr, int n);

The function accepts an integer array 'arr' and its length 'n'. Implement the function to find and return the number of superior elements in array 'arr'

Assumptions:

1. N > 0.
2. Array index starts from 0.

Input:
7 9 5 2 8 7
Output:
3

Alex Gives You a positive Number N and wants you to rearrange the bits of the number in its binary representation such that all set bits are in consecutive order. Your task is to find and return an integer value representing the minimum possible number that can be formed after re-arranging the bits of the number N.

Input 1:

10

Output 1:

3

Input 2:

2

Output 2:

2

Find count of magical numbers from 1 to N
A number is magical if:
    Convert to binary.
    Replace 0 with 1 and 1 with 2 in binary string.
    Claculate sum of all diits in binary string.
    Resultant must be an odd number

Input:
5
Output:
2
Explanation:
  1 -> Binary = 1 -> convert to = 2   sum = 2 (even)
   2 -> Binary = 10 -> convert to = 21 sum = 3 (odd)
   3 -> Binary = 11 -> convert to = 22 sum = 4 (even)
   4 -> Binary = 100 -> convert to = 211 sum = 4 (even)
   5 -> Binary = 101 -> convert to = 212 sum = 5 (odd)

You are provided with a string which has a sequence of 1s and 0s. This sequence is the encoded version of a english word. You are supposed to write a program to decode the provided string and find the original word. Each uppercase Alphabet is representing by a sequence of 1s

Note:

Alphabet which is represented by a sequence of 1s will get separate in between by single zero.

Input 1:
10110111
Output 1:
ABC
Explanation:
 1 0 1 1 0 1 1 1
  A   B   C
Input 2:
101101
Output 2:
ABA
Input 3:
101101111
Output 3:
ABD

A stream of n data packets arrives at a server. This server can only process packets that are exactly 2^n units long for some non-negative integer value of n (0<=n).

All packets are repackaged in order to the 1 largest possible value of 2^n units. The remaining portion of the packet is added to the next arriving packet before it is repackaged. Find the size of the largest repackaged packet in the given stream.

Constraints:

1<=n<=10^5

1<=arriving Packets[i] size<=10^9

Input 1:
5
12 25 10 2 8
Output 1:
16
Example:
arriving Packets = [12, 25, 10, 7, 8]
The first packet has 12 units. The maximum value of 2^n that can be made has $2^n = 2^3 = 8$ units because the next size up is $2^n = 2^4 = 16$ (16 is greater than 12).
12 – 8 = 4 units are added to the next packet. There are 4 + 25 = 29 units to repackage, $2^n = 2^4 = 16$ is the new size leaving 9 units (29-16 = 9)
Next packet is 9 + 10 = 29 unists & the maximum units(in 2^n) is 16 leaving 3 units.
3 + 7 = 10 , the max units is 8 Leaving 2 units, and so on.
The maximum repackaged size is 16 units.

THANK YOU