September 1, 2025

CONTEST 3 DISCUSSION

Ques 1. Count Proper Words

A proper word is a string made of lowercase English alphabets and contains at least 1 vowel and at least 1 consonant.

You need to find how many proper words of length *N* exist. As, the number can be very large, print it as modulus *1000000007*.

Note - a,e,i,o,u are considered as vowels, and the rest of the alphabets are considered as consonants.

Valid String contains: atleast 1 vowel & 1 consonant.

S = "BCD" No
S = "A" No
S = "ABCE" YES

What will be the number of valid strings with length = 1
⇒ 0

→ To make n length string using recursion (print)

```
int cnt = 0;
bool valid(string s)
{
    bool vowel = 0, consonent = 0;
    for (auto i : s)
    {
        if (isVowel(i))vowel = 1;
        if (isConsonent(i))consonent = 1;
    }
    if (vowel and consonent)return 1;
    return 0;
}
void rec(int i, string s, bool vowel, bool consonent)//n*(26^n)
{
    //base case
    if (i == n)
    {
        if (vowel and consonent)
        {
            cnt++;
            cout << s << endl;
        }
        return;
    }
    for (char c = 'a'; c <= 'z'; c++)
    {
        bool tempvowel = vowel, tempconsonent = consonent;
        if (isVowel(c))tempvowel = 1;
        else tempconsonent = 1;
        rec(i + 1, s + c, tempvowel, tempconsonent);
    }
```

→ Now, we have to count
All consonant = 21^N
All vowels = 5^N
Total = 26^N

atleast 1 cosnonant & 1 vowel:
⇒ Total − (all cosnonant + all vowel)
$= 26^N - (21^N - 5^N)$

Ques 2. Special Triangle

A triangle of numbers is called a 'Special Triangle' **if every number in each row (except the last row) is strictly less than the two numbers which are just below it.**

Consider the example shown in the first figure. Every number is less than the numbers which are just below it (except the numbers in the last row). For the first row, the numbers which are just below the 2 are 3 and 3. For the second row, the elements below the first 3 are 5 and 4, and for the second 3 are 4 and 5.

```
    2
  3   3
5   4   5
```

Now consider the second example figure. The element marked in red is violating the property of Special Triangle.

```
    2
 [5]   3
5   4   6
```

You have given an array of numbers. You have to find whether you can construct a Special Triangle using only those numbers. If you can construct such a triangle, print 'YES', otherwise 'NO'.

Special Triangle:

a

a<b　　　　a<c ← have to make sure

b　　　　c

b<d, b<e, b<f　　c<d, c<e, c<f

d　　e　　f

Sort the given list of array. Then, we will start building the triangle using smallest.

Ques.3. GooGol 1

Given array A of N integers $A_1, A_2, A_3, \ldots, A_N$. You are also given two integers S and M.

You can perform at most M operations

- In each operation you can pick a subarray of size S and increment the value of each element in that subarray by 1.

Apply the operations in an optimal way such that the **minimum value in array A is maximum**. Print the maximum possible value of the minimum element in array A.

S = 3 (subarray size)
M = 5 (Operations)

[1 2 3 2 1 4]
increment M = 5 − 1 = 4
[2 3 4 2 1 4]
M = 3
[2 3 4 3 2 5]
M = 2
[3 4 5 3 2 5]
M = 1
[3 4 5 4 3 6]
M = 0 (stop)
⇒ [4 5 6 4 3 6]
Ans
Since it is min

Brute Force:

1<3, Yes check → 3

[1 2 3 2 1 4]
+2 +2 +2

[3 4 5 2 1 4]
4<3, No

[3 4 5 2 1 4]
5<3, No

+1 +1 +1
[3 4 5 2 1 4]
2<3, Yes
[3 4 5 3 2 5]
+1
2<3, Yes
+1
[3 4 5 3 3 6]
6<3, No

Total Steps = 2+1+1 = 4

Check(x)

```
{ int step=0;
  for(i=0; i<n; i++)
  {   if(a[i]< x)
      {
          int add= x-a[i]; step+=add;
          for(j=i; j<i+s; j++)
          {
              a[j]+=add;
          } } }
  if(step <= M) return 1
  else return 0 .
```

T.C = $O(N^2)$

Optimise the second loop for linear T.C.

Approach: Partial Sum

1	3	3	3	1	0
+1	+2			−2	−1
0	0	0	0	0	0
0	1	2	3	4	5

[1 → 3] → +2, add +2 at 1, −2 at (3+1)
[0 → 4] → +1

Maintain Prefix Sum instead of the second for loop.

```
nt check(int x)
{
    vector<int>partial(n,0);
    for(int i=0;i<n;i++)
    {
        if(i>0)partial[i]+=partial[i-1];
        a[i]+=partial[i];
        if(a[i]<x)
        {
            int add=x-a[i];
            partial[i]+=add;
            if(i+m<n)partial[i+m]-=add;
            a[i]+=add;
        }
    }
```

Ques 4. GooGol 2

Given an array A of N integers $A_1, A_2, \ldots, A_N$. You are also given two integers S and M.

- You can perform at most M operations and in each operation :
    - you can pick a **subset** of indices of size S, and increment the value of each chosen element by 1.
    - Apply the operations in an optimal way such that the minimum value in array A is maximized.

Print the **Maximum** possible value of the minimum element in array A.

Subarrays → Two pointers, Sliding window

Subsets → sort

→ Sort

→ if i can be increased then can easily go till its skipping the second for loop.

Ques 5. People on Number line.

There are X people that will stand on a 1D number line.

There are Y mutually disjoint intervals along the number line on which they can stand. Multiple people can stand on a single interval.

The people want to stand at distinct integer points within those intervals to maximize Z, where Z is the distance between the closest pair of people. Determine the largest possible value of Z.

Intervals: 5 → people

[1 – 4]

[6 – 12]

[19 – 20]

→ Person 1

P_1 P_2 P_3 P_4 P_5 ← Persons

$1 \underbrace{\quad}_{3} 4 \underbrace{\quad}_{2} 6 \underbrace{\quad}_{4} 10 \underbrace{\quad}_{9} 19$ ← Placement

Z (closest pair distance) = 2 (Ans)

Maximise the Z.