Computer Engineering Dept. **Design & Analysis of Algorithms**
Cairo University
Fall 2017

Problem A

the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. given two binary strings A,B with length n . we need to find string C such that Hamming distance between A,C is equal to Hamming distance between B,C

Input

The first line contains string A of length n.

The second line contains string B of length n.

The length of string n is within range from 1 to 10^5.

Output

Print “YES” if solution exist . If no such string exist, print on a single line "NO" (without the quotes).

Inputs	Outputs
0001 1011	YES
000 111	NO

Problem B

King Arthur wants to count number of digits from 1 to n. for example if n=13 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, which is 17 digits he could not solve the problem for very large “n” so he asked you for help.

Input test file contain serveral test case one line contains integer n ($1 \le n \le$ 10^9) **Output**

Print the number of digits.

inputs	outputs
13	17
55	101
4	4

Problem C

we are in a shop . We are offered different types of products (for example, 3 shirts, 2 belts, 4 shoes, ...). We have to buy **exactly** one from each product (ex : one shirt , one belt and one shoe …..), Such that we spend as much money as possible.

Inputs

- first line is N , number of test cases
- each test case start by M,C where “M” is the available budget with you. “C” is the number of available products. (ex: if we have shirts , belts , shoes then C = 3) (1 ≤ M ≤ 200, and 1 ≤ C ≤ 20)
- for the next C lines , the first integer, K (1 ≤ K ≤ 20), indicates the number of different types for each product and it is followed by K integers indicating the price of each one.

Output

for each test case, the output should consist of one integer indicating the maximum amount of money necessary to buy one element of each product without exceeding the initial amount of money. If there is no solution, you must print ‘no solution’.

Inputs	outputs

3
100 4
3 8 6 4
2 5 10
4 1 3 3 7
4 50 14 23 8
20 3
3 4 6 8
2 5 10
4 1 3 5 5
5 3
3 6 4 8
2 10 6
4 7 3 1 7
	75
	19
	no solution

Problem 1-1. [15 points] **Asymptotic Practice**

For each group of functions, sort the functions in increasing order of asymptotic (big-O) complexity:

(a) [5 points] **Group 1:**

$$\begin{aligned} f_1(n) &= n^{0.999999}\log n \\ f_2(n) &= 10000000n \\ f_3(n) &= 1.000001^n \\ f_4(n) &= n^2 \end{aligned}$$

(b) [5 points] **Group 2:**

$$\begin{aligned} f_1(n) &= 2^{2^{1000000}} \\ f_2(n) &= 2^{100000n} \\ f_3(n) &= \binom{n}{2} \\ f_4(n) &= n\sqrt{n} \end{aligned}$$

(c) [5 points] **Group 3:**

$$\begin{aligned} f_1(n) &= n^{\sqrt{n}} \\ f_2(n) &= 2^n \\ f_3(n) &= n^{10}\cdot 2^{n/2} \\ f_4(n) &= \sum_{i=1}^{n}(i+1) \end{aligned}$$