Course Name: Algorithm Design and Analysis
Course ID: 180086081200P1002H

Chapter Name: Greedy

For problems 1-5, you should do at least the following things:

1. Describe your algorithm in natural language AND pseudo-code;
2. Describe the optimal substructure and DP equation;
3. Prove the correctness of your algorithm;
4. Analyse the complexity of your algorithm.

---

1. Commando War

   There is a war and it doesn't look very promising for your country. Now it's time to act. You have a commando squad at your disposal and planning an ambush on an important enemy camp located nearby. You have N soldiers in your squad. In your master-plan, every single soldier has a unique responsibility and you don't want any of your soldier to know the plan for other soldiers so that everyone can focus on his task only. In order to enforce this, you brief every individual soldier about his tasks separately and just before sending him to the battlefield. You know that every single soldier needs a certain amount of time to execute his job. You also know very clearly how much time you need to brief every single soldier. Being anxious to finish the total operation as soon as possible, you need to find an order of briefing your soldiers that will minimize the time necessary for all the soldiers to complete their tasks. You may assume that, no soldier has a plan that depends on the tasks of his fellows. In other words, once a soldier begins a task, he can finish it without the necessity of pausing in between.

   Input: There will be multiple test cases in the input file. Every test case starts with an integer $N(1 <= N <= 1000)$, denoting the number of soldiers. Each of the following N lines describe a soldier with two integers $B(1 <= B <= 10000)\&J(1 <= J <= 10000)$. B seconds are needed to brief the soldier while completing his job needs J seconds. The end of input will be denoted by a case with $N = 0$. This case should not be processed.

   Output: For each test case, print a line in the format, "Case X: Y", where X is the case number $\&Y$ is the total number of seconds counted from the start of your first briefing till the completion of all jobs.

2. DNA Consensus String

   The Hamming distance is the number of different characters at each position from two strings of equal length. For example, assume we are given the two strings "AGCAT" and "GGAAT." The Hamming distance of these two strings is 2 because the 1st and the 3rd characters of the two strings are different. Using the Hamming distance, we can define a representative string for a set of multiple strings of equal length. Given a set of strings $S = \{s_1, ..., s_m\}$ of length n, the consensus error between a string y of length n and the set S is the sum of the Hamming distances between y and each si in S. If the consensus error between y and S is the minimum among all possible strings y of length n, y is called a consensus string of S. For example, given the three strings "AGCAT"

"AGACT" and "GGAAT" the consensus string of the given strings is "AGAAT" because the sum of the Hamming distances between "AGAAT" and the three strings is 3 which is minimal. (In this case, the consensus string is unique, but in general, there can be more than one consensus string.) We use the consensus string as a representative of the DNA sequence. For the example of Figure 1 above, a consensus string of gene X is "GCAAATGGCTGTGCA" and the consensus error is 7.

	DNA sequence of gene X
Cat:	**GCATATGGCTGTGCA**
Dog:	**GCAAATGGCTGTGCA**
Horse:	**GCTAATGGGTGTCCA**
Cow:	**GCAAATGGCTGTGCA**
Monkey:	**GCAAATCGGTGAGCA**

Figure 1: DNA sequences of gene X in five animals.

Input: Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing two integers m and n which are separated by a single space. The integer $m(4 \leq m \leq 50)$ represents the number of DNA sequences and $n(4 \leq n \leq 1000)$ represents the length of the DNA sequences, respectively. In each of the next m lines, each DNA sequence is given.

Output: Your program is to write to standard output. Print the consensus string in the first line of each case and the consensus error in the second line of each case. If there exists more than one consensus string, print the lexicographically smallest consensus string.

3. Opponents

   Arya has n opponents in the school. Each day he will fight with all opponents who are present this day. His opponents have some fighting plan that guarantees they will win, but implementing this plan requires presence of them all. That means if one day at least one of Arya's opponents is absent at the school, then Arya will beat all present opponents. Otherwise, if all opponents are present, then they will beat Arya.

   For each opponent Arya knows his schedule — whether or not he is going to present on each particular day. Tell him the maximum number of consecutive days that he will beat all present opponents.

   Note, that if some day there are no opponents present, Arya still considers he beats all the present opponents.

   Input: The first line of the input contains two integers n and d ($1 <= n, d <= 100$) — the number of opponents and the number of days, respectively. The i -th of the following d lines contains a string of length n consisting of characters '0' and '1'. The j -th character of this string is '0' if the j -th opponent is going to be absent on the i -th day.

   Output: Print the only integer — the maximum number of consecutive days that Arya will beat all present opponents.

4. Minimum Varied Number

   Find the minimum number with the given sum of digits s such that all digits in it are distinct (i.e. all digits are unique).

   For example, if $s = 20$, then the answer is 389 . This is the minimum number in which all digits are different and the sum of the digits is 20 ($3 + 8 + 9 = 20$).

   For the given s print the required number.

Input: The first line contains an integer t ($1 \le t \le 45$) — the number of test cases. Each test case is specified by a line that contains the only integer s ($1 \le s \le 45$).

Output: Print t integers — the answers to the given test cases.

5. Joey Takes Money

Joey is low on money. His friend Chandler wants to lend Joey some money, but can't give him directly, as Joey is too proud of himself to accept it. So, in order to trick him, Chandler asks Joey to play a game.

In this game, Chandler gives Joey an array $a_1, a_2, \ldots, a_n$ ($n \ge 2$) of positive integers ($a_i \ge 1$).

Joey can perform the following operation on the array any number of times:

1. Take two indices i and j ($1 \le i < j \le n$) . 2. Choose two integers x and y ($x, y \ge 1$) such that $x \cdot y = a_i \cdot a_j$. 3. Replace a_i by x and a_j by y .

In the end, Joey will get the money equal to the sum of elements of the final array.

Find the maximum amount of money ans Joey can get but print $2022 \cdot \text{ans}$. Why multiplied by 2022 ? Because we are never gonna see it again!

It is guaranteed that the product of all the elements of the array a doesn't exceed 10^{12} .

Input: Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 4000$). Description of the test cases follows. The first line of each test case contains a single integer n ($2 \le n \le 50$) — the length of the array a . The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the array itself. It's guaranteed that the product of all a_i doesn't exceed 10^{12} (i. e. $a_1 \cdot a_2 \cdot \ldots \cdot a_n \le 10^{12}$).

Output: For each test case, print the maximum money Joey can get multiplied by 2022 .