TeamsCode Summer 2020 Online Coding Contest
Advanced Problem Set

Rules:

1. You have 3 hours to solve as many of the 15 problems as you can. Each problem has 3 test cases and each correct test case is worth 10, 20, or 30 points, depending on whether the problem is easy, medium, or hard, respectively.
2. Email your solution code to **teamscodeofficial@gmail.com**. You get 2 submission attempts for each problem. In the subject line, please specify your **Division, Team Name, Problem #, and Language** (e.g. Advanced, Team A, Easy #3, Java). In the message body, attach a text file with your code (**Strongly Recommended**) or copy-paste your code in its entirety.
3. In the case of a tie, the winner will be the team that made their last correct submission first.
4. You must use standard console input to read in the test cases and output your answers. Specifically, Java uses `Scanner(System.in)`, Python uses `input()`, C++ uses `cin >>`, and C# uses `Console.OpenStandardInput()`. Likewise, print your answer to the console using standard output.
5. Link to the full logistics can be found here.

Table of Contents

Scared Skiing [Easy #1]

You and your friends decide to go skiing for summer break! However, this is your first time going skiing, so you are concerned about how steep the slope is. You are given the coordinates of the top of the slope (x_1, y_1) and the bottom of the slope (x_2, y_2). The x values represent the horizontal positions, whereas the y values represent the vertical positions. Determine how steep the slope is, the vertical displacement over the horizontal displacement. Make sure to round your answer down as an integer. Note: y_2 will always be less than y_1.

Input:

The first line contains 4 integers. The first 2 represent the top of the slope (x_1, y_1) and the second 2 represent the bottom of the slope (x_2, y_2), where $(1 \leq x_1, y_1, x_2, y_2 \leq 1,000)$.

Output:

Print the slope given the coordinates of the two points as a truncated integer.

Sample Input:

```
10 10 8 3
```

Sample Output:

```
3
```

The output is 3, since the change in y is 7 and the change in x is 2.

Perfect Promposal [Easy #2]

Prom is in a week and you still haven't made a poster to ask your crush to the dance! Every day you delay asking, your crush is less likely to say yes. However, delaying allows you to make your poster higher quality. You know that your crush will go with you if the poster quality Q_i multiplied by the number of days remaining is at least a certain constant M. Determine how many days of the week your crush will say yes.

Input:

The first line contains integer $1 \leq M \leq 1,000$.
The second line contains 7 integers $1 \leq Q_i \leq 1,000$.

Output:

Print the number of days your promposal will be successful.

Sample Input: (successful days highlighted for convenience)

```
40
3 7 7 14 15 18 20
```

Sample Output:

```
3
```

3 days before prom, the value will be 15 * 3 = 45 > 40, thus you'll get a yes. On the other hand, 1 day before prom the value will be 20 * 1 = 20 < 40, so you'll get a no.

Tricky Travels [Easy #3]

You're playing a solo board game called *Tricky Travels* where you travel around N tiles placed in a circular formation. Every tile has a number T_i that indicates that for the next turn, you will move T_i spaces around the circle. If you move past the last N^{th} tile, you will loop back to the first tile. The game ends when you land on the finish tile. Print the number of moves it takes to reach the space represented by a 0. If the finish tile is never reached, print -1.

Input:

The first line contains an integer $1 \leq N \leq 20$, the number of spaces.
The second line contains N numbers $0 \leq T_i \leq N$, which defines the board.
The finish is represented by a 0.

Output:

Print the number of moves it takes to reach the finish tile. If the finish tile is never reached, print -1.

Sample Input (path is highlighted for convenience):

```
5
2 0 1 3 5
```

Sample Output:

```
3
```

Essay Expert [Easy #4]

Your friend is an excellent writer, but he is sloppy at typing; he often repeats words unintentionally. Your job is to go through his essay and remove words that are repeated. Note: no word will occur more than two times in a row.

Input:

The first line contains an integer $1 \leq N \leq 50$, the number of words in your friends essay. The next line contains N words.

Output:

Output the edited String that omits repeated words in one line.

Sample Input:

```
9
I often repeat repeat words in my my essays
```

Sample Output:

```
I often repeat words in my essays
```

Wilderness Wasps [Easy #5]

Oh no! Wilderness Willy was exploring the wilderness when a group of wasps escaped their nest and began chasing after him! One by one, from least venomous to most venomous, the wasps sting Wilderness Willy. Each of the N wasps injects a certain amount of venom k_i into Willy individually.

When there are x wasps sting Willy at the same time, each of these wasps do x times more damage than they would individually. For example, If 3 wasps were stinging him at the same time, each wasp would do triple the damage. Wilderness Willy can take up to V venom without going to the hospital. Given a list of how much damage each individual wasp does, compute the maximum number of wasps that he can get stung by simultaneously without having to go to the hospital (if he can get stung by all N simultaneously, print N).

Input:

The first line contains two integers $1 \leq V \leq 10,000$ and $1 \leq N \leq 100$, Willy's venom threshold and the number of wasps, respectively. The next line contains N space separated integers $1 \leq k_i \leq 500$, the amount of damage each wasp deals on its own.

Output:

Output the maximum number of wasps Willy can get stung by without having to go to the hospital.

Sample Input:

```
100 5
4 9 11 13 17
```

Sample Output:

```
3
```

Quarantine Qualms [Medium #1]

After being in quarantine for several months, you are bored out of your mind. You decide to go outside to get some fresh air. The probability that you contract COVID-19 from anyone else outside during any given hour is 0.02, or 2%. For instance, if you are outside with two other people for one hour, the probability that you contract COVID-19 from either of them is 0.0396, which is about 4%. Note that each probability is independent of the number of people and hours that you are out, meaning the probability of contracting COVID-19 from one person who is out for 2 hours is the same as that of two people who are out for 1 hour. Given the times in a given day that you and others are outside, compute the probability that you contract COVID-19.

Input:

The first line contains two integers: the hour you first walk outside, t_1, and the hour that you go back inside, t_2. ($0 \leq t_1 \leq t_2 \leq 24$). The next line contains the number of other people $0 \leq N \leq 100$. In the next N lines, you are given the hours that people first walk outside and come back inside.

Output:

Output as a percentage rounded to the nearest integer, the probability that you contract COVID-19.

Sample Input:

```
2 16
3
15 24
14 16
19 22
```

Sample Output:

```
6
```

Sleepy Spider [Medium #2]

You're just about to go to bed after a long day of solving coding problems when you suddenly realize that your house has become infested with spiders! There are spiders on all 6 faces of your room, including all 4 walls, the ceiling, and the floor. Each face is given an index from 0 to 5. Suppose on a certain day, there are k_j spiders on the j^{th} face of your room. If there are less spiders on the $(j+1)^{th}$ face than the j^{th} face, then half of the spiders (rounded down if odd) on the j^{th} face will move to the $(j+1)^{th}$ face. All spiders move at the same time and spiders on the 5th face will move to the 0th face. After N days, print the number of spiders on each face of your room, from 0 through 5.

Input:

The first line contains the integer $1 \leq N \leq 100$. The next line contains 6 space-separated integers $0 \leq k_j \leq 1000$ denoting the number of spiders on the faces 0 to 5, in that order.

Output:

Print the number of spiders on each face from 0 to 5, one on each line, in that order.

Sample Input:

```
2
1 2 3 4 5 6
```

Sample Output:

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

Europe Escapade [Medium #3]

This summer, you and your N friends are handed an list of K_i European countries. Since you are on a tight schedule, you only get one day to visit each country and have to go on a train ride to the next country the next day. Each country is connected by exactly one railroad, so there's only one way to get from one country to another.

Although the sequence of countries you and your friends visit are different from each other, you will still be able to see each other on connecting trains. In order to maximize the time you see each other, you and your friends must decide which day you begin your trip. Compute the maximum number of train rides you can share with your friends by altering the day that each of you leave. Don't worry about train rides your friends share among each other.

Input:

The first line contains an integer $1 \leq N \leq 10,000$, the number of friends (excluding you) that are participating on the trip. The next line contains $N+1$ integers, the number of countries you're traveling to, followed by K_1 to K_N, detailing how many countries that your friends are traveling to. The next $N+1$ lines contain K_i capital letters each, describing which order the countries are traveled to for that friend.

Output:

Output the maximum number of train rides you can share with friends.

Sample Input (shared train rides bolded for convenience):

```
4
5 3 5 4 4
A C B F D
C B A
A C E F D
E B F D
A C F D
```

Sample Output:

```
11
```

* The train ride from F-D on the last friend can’t be shared because if A-C is shared, you travel to an intermediate country, while your friend does not, therefore causing the rides to be offset. *

Paragliding Pioneer [Medium #4]

As part of a dare, you paraglide off the Yosemite Half Dome. Unsure of where to land, you remember that earlier that day, you had a talk with N scientists who each told you to land within a certain range between L_i and U_i meters (denoting the lower and upper bounds) from the base of the cliff, including L_i and U_i. In order to have the safest landing, you want to land x meters away, where x is equal to the integer that is in the largest number of scientist's ranges. If multiple numbers are applicable, choose the smallest such x, since you want to land near the half dome.

Input:

The first line contains an integer $2 \leq N \leq 1000$. The next N lines each contain two space-separated integers $0 \leq L_i \leq 10^6$ and $0 \leq U_i \leq 10^6$.

Output:

Output the integer that appears in the most scientists' ranges. If there are multiple integers, choose the lowest of the set.

Sample Input:

```
4
2 8
7 15
1 4
5 10
```

Sample Output:

```
7
```

Courageous Colonists [Medium #5]

After winning the TeamsCode raffle you decide to spend your winnings on a trip to the midwest United States. You make up your mind to stay and after years of grueling labor become a well known colonist. You have a certain amount N_i of sheep, wood, brick, ore, and wheat, and you want to allocate these resources in the best way possible – so that you earn the most "points." In order to do this, you build a chain of settlements and cities connected by a sequence of roads in one straight line. The structures and rules are as follows:

- A settlement is worth one point. It costs one sheep, one wood, one brick, and one wheat to build.
- A road is worth zero points, but is required for connecting settlements. It costs one wood and one brick to build.
- A city can be built to replace a settlement and gives one additional point. It costs two wheat and three ore to build.
- At the cost of three ore, three wheat, and three sheep, you may build a town hall to manage your colony. This nets you two points, but you may only build one town hall. This does not need to be built along roads, nor does it replace settlements or cities.
- A settlement cannot be built unless it is at least a distance of two roads from adjacent settlements and cities. Assume that all roads are built in one straight line for simplicity.

Input:

The first line contains 5 integers $0 \leq N_i \leq 10^5$, indicating the amount of resources you start with. It is in the order sheep, wood, brick, ore, and wheat.

Output:

Output the largest number of points you can form given your resources.

Sample Input:

```
3 7 7 9 10
```

Sample Output:

```
6
```

Output Details: You can earn 6 points by building 3 settlements,upgrading them to 3 cities, and connecting them via 4 roads.

C -- C -- C

Lucrative Lootcrates [Hard #1]

You are investing your allowance money into the lucrative craft of lootcrate trading in a popular video game in hopes of making a quick buck. Each day, you may buy one loot crate, sell a lootcrate (if you have at least one), or do nothing. On the first day, you start off with zero loot crates. Find the minimum number of transactions (buying or selling) needed to make a profit of at least P (a decimal value), given the price (a decimal value) for a given lootcrate on each of N days.

Input:

The first line contains 2 integers $2 \leq N \leq 20$ and $5 \leq P \leq 100$. The next line contains N space-separated decimal values representing the prices of the loot crates on each day.

Output:

Output the minimum number of transactions needed to make a profit of at least P in the first N days.

Sample Input: (Transaction days are bolded for convenience):

```
6 5.5
5.6 9 11.2 8.3 10.9 11.1
```

Sample Output:

```
2
```

Vacuum Voyager [Hard #2]

A cleaning robot has been tasked with vacuuming up dust as it travels from the northwestern (NW) corner of a building to the southeastern (SE) corner of a building, given limited energy C. The building's layout is an $N \times N$ grid. Each room in this grid is either empty, contains either dust to clean up, or contains a charging station for the robot. If the room contains dust to clean up, it is marked with a negative number $E_{i,j}$, the energy cost needed to remove the dust in the room upon entering. If the room has a charging station, it is marked with a positive number $E_{i,j}$, the energy restored to the robot's battery upon entering the room. The robot must make its way from the NW room in the grid to the SE room in the grid without running out of battery. $E_{i,j}$ in the NW and SE corners will always be 0. For each room, it can only move south, or east, not backwards. Please output the maximum energy the robot can have when it reaches the southeast corner.

Input:

The first line contains two space-separated integers $2 \leq N \leq 50$ and $0 \leq C \leq 1,000$. The next N lines contains N space-separated integers $-9 \leq E_{i,j} \leq 9$ each.

Output:

Output the maximum energy the robot can have when it reaches the southeast corner.

Sample Input (path bolded for convenience):

```
4 5
0 1 0 2
-2 -1 1 1
-3 -6 2 1
2 0 -1 0
```

Sample Output:

```
10
```

Nanovirus Neutralizers [Hard #3]

A rogue, self-replicating nanovirus has escaped containment and appeared in the country! For the last t days, these nanoviruses have been spreading outwards from their starting point. Each day, the nanovirus spreads outwards one unit up, down, left, or right in any of these directions possible. The nanovirus cannot spread diagonally. Assume that the country is an $N \times N$ grid, and the nanovirus starting point is denoted by the character 'Z'. There are a few barriers already in place, denoted by the character '#', that help contain the virus. It is your job to build new barriers to contain these nanoviruses and prevent any further spread. You must build the barriers in such a way that the nanoviruses cannot spread to even a single un-contaminated grid location.

Input:

The first line contains the integer $2 \leq N \leq 30$, followed by the integer $1 \leq t \leq 10$
The next N lines contain N characters each. Ground zero is represented by Z, and pre-existing barriers are represented by a hashtag(#), and empty spaces are represented by an asterisk (*).

Output:

Output the number of barriers, not including pre-existing ones, needed to encapsulate locations affected by the virus. Assume the edges of the country are surrounded by barriers.

Sample Input: (necessary blockades bolded for convenience)

```
8 3
*****#
**#*****
*****#***
**#*Z***
**#****#*
******#**
*******
##*****
```

Sample Output:

```
11
```

Prime Pillager [Hard #4]

Arthur is an avid mathematician in search of prime numbers. As part of his hunt for primes, he randomly generates a string S of random digits of length L and wants to find primes in it. To make this task more exciting, he decides he wants to find the smallest number of partitions he can make to cut the strings into separate numbers such that no partitioned number exceeds 7 digits, and every number is a prime. Print the minimum number of separate numbers the string can be split into while still satisfying the above two constraints. If no such partitioning is possible, print -1.

Input:

The first line contains the string S of length $1 \leq L \leq 10^5$.

Output:

Print the number of segments the string is split into.

Sample Input:

```
43917291
```

Sample Output:

```
2
```

In this case, the primes are 439 and 17291. Thus, there are 2 separate numbers.

Juice Jam [Hard #5]

At a post-quarantine party, you are trying to fill up a large pitcher to the brim with juice by pouring in juice from various juice boxes of different sizes. You have a limited number of juice boxes of each size, but would like to find out how many ways you can fill up the pitcher exactly. You have N different sizes of juice boxes. For each of these sizes, you have k_i juice boxes correspondingly containing V_i milliliters of juice each. The volumes of the juice boxes are listed in descending order. You are trying to find the number of ways to fill a pitcher of volume S milliliters.

Input:

The first line contains 2 space-separated integers $2 \leq N \leq 15$ and $2 \leq S \leq 300$. The next N lines each contain 2 space-separated integers $1 \leq k_i \leq 300$ and $1 \leq V_i \leq 200$.

Output:

Output the number of different ways you can fill the pitcher of volume S.

Sample Input:

```
3 13
1 10
4 5
3 1
```

Sample Output:

```
2
```

In this case, you must use all 3 of the 1-mL boxes. Thus, the problem becomes filling a 10 mL pitcher using 5-mL and 10-mL boxes. There are 2 ways to do this. You could combine a 10-mL box and three 3-mL boxes to make 13 mL of juice, or two 5-mL boxes and one 3-mL box to make 13 mL of juice.