Fall 2020 mBIT Standard Division

November 14, 2020

These problems are roughly ordered by difficulty. However, you should read and think about as many problems as you can in the time given. Good luck and happy coding!

Contents

Thanks to Evan Chen for letting us use his style file

Program Specifications

The memory limit for every problem is 256 MB.

Time Limits (seconds)			
Problem	**C++**	**Java**	**Python**
Apple Pie	1	1	1
Double Trouble	1	1	1
Explorers	1	1	2
Banquet	1	1	1
Climbing Trees	1	1	1
Weights	1	1	1
Stone Piles	1	1	2
Number Game	1	1	1
Heating Rocks	1	1	1
Calendars	1	1	1
Cathedral	1	3	3
Gemstones	2	2	6

Advice

Look at the pretests. You can access all 10 pretests for each problem once you've made a submission. *Some of the pretests are reduced in size to help you debug your program.* Keep in mind that your final submission will be judged on a separate set of 40 hidden system tests for the official rankings.

Take advantage of subtasks. Many problems have subtasks which allow you to earn points for solving easier cases. Each subtask guarantees that a certain proportion of the system tests satisfies some additional constraint. Remember, you will get one point for each system test you pass, **plus 20 points** if you get all 40 system tests correct.

Watch out for integer overflow. When a problem uses large values, make sure you use `long` (in Java) or `long long` (in C++). Python integers cannot overflow.

Consider using fast I/O. For problems with large input sizes, you may want to use faster I/O methods to prevent a time limit error. Here is how to use fast I/O in each language:

- In Python, write `from sys import stdin, stdout` at the top of your program. When reading input, use `stdin.readline()`. To write output, use `stdout.write()`.
- In Java, use a custom Scanner class as shown here.
- In C++, write `ios_base::sync_with_stdio(false); cin.tie(NULL);` at the top of your `main` method. Then you can use `cin` and `cout` as usual. Printing a single newline character (`\n`) is faster than `endl`, which flushes the output stream (`endl` is only needed in interactive problems.)

Print extra digits for non-integer values in C++. If you are printing a `double` value in C++, by default it will only output a few digits (which may result in a wrong answer from our grader). To output real values with more precision, write `cout << setprecision(16);` at the top of your program. Our grader will accept real values in fixed **or** scientific notation (so `1234.56789`, `1.23456789E3`, and `1.23456789e+003` are treated the same). There will always be a tolerance for small relative errors between your solution and the correct answer.

Special considerations for Python. In Python, avoid multidimensional lists. Nesting lists may slow your program down. You can use a trick known as array flattening instead. Additionally, if you are coding in Python, consider submitting your solution in PyPy. It behaves in the same way as Python, but is often faster (the time limits for PyPy are the same as Python.) Putting your code in a `main` method can speed up run time as well. Finally, **do not** use `exit()`.

Ask for clarifications! If you are confused about a problem statement, do not hesitate to message us.

"Time is the longest distance between two places."
—Tennessee Williams

"We all have our time machines. Some take us back, they're called memories. Some take us forward, they're called dreams."
—Jeremy Irons

"Time limit exceeded on pretest 3."
—The grader

§1 Apple Pie

The first Thanksgiving celebrations featured lobster and seal, prepared using traditional Native American spices and cooking methods.

Claire the Pilgrim is planning to bake K apple pies for her town's Thanksgiving celebration. The recipe she is using requires 10 apples for each pie. In the market, apples are sold in baskets containing exactly M apples each. If Claire already has N apples, how many baskets does she need to buy so that she can bake K pies?

Input Format:

The only line of input contains K, M, and N ($1 \leq K, M, N \leq 1000$).

Output Format:

Output one line with the minimum number of baskets Claire needs to buy to have enough apples for K pies. If she already has enough apples, the answer is 0.

Sample Input:

```
10 8 73
```

Sample Output:

```
4
```

Claire needs a total $10 \cdot 10 = 100$ apples for the pies. She needs $100 - 73 = 27$ more apples, so she must buy 4 baskets.

§2 Double Trouble

The earliest alphabet was created by the Phoenecians 3000 years ago. Its characters were influenced by Egyptian Hieroglyphs.

Phil the Phoenician is editing a book about his adventures on the Mediterranean Sea. For some reason, Phil doesn't want it to have any double letters. He plans to go through his book word by word and find all of the times a letter is repeated twice in a row. At each of these places, he will remove one of the repeated letters. For example, if the original text is `hmmmm hello otter`, Phil will change it to `hmmmm helo oter`. **If a letter is repeated three or more times, Phil ignores it**. Also, letters separated by a space are **not** considered adjacent.

Phil will give you an excerpt of his book. Figure out what the text will look like after Phil has edited it.

Input Format:

The input contains a single line of text S ($1 \leq |S| \leq 10^4$). All characters will be lowercase letters or spaces.

Output Format:

Output the text with all double letters converted into single letters.

Sample Input:

```
ugghhh i hate bookkeepers so much
```

Sample Output:

```
ughhh i hate bokepers so much
```

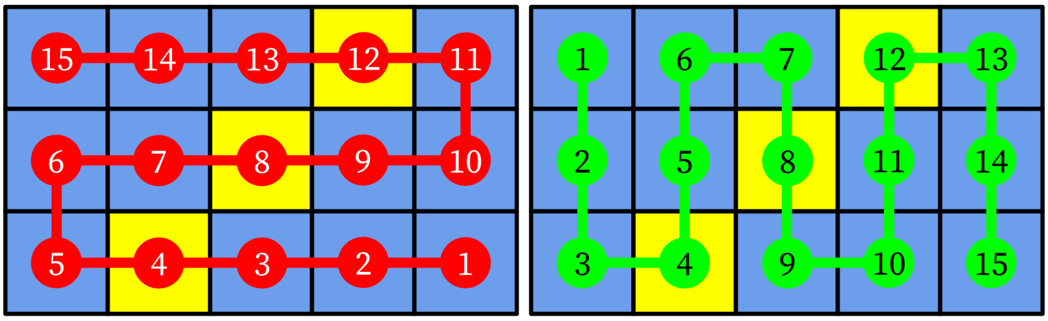
The double letters `gg`, `oo`, `kk`, and `ee` have been reduced to single letters, but the `hhh` is unchanged since it is a sequence of three letters.

§3 Explorers

Zheng He was a Chinese mariner during the Ming dynasty whose voyages spanned Southeast Asia, the Indian subcontinent, and East Africa.

Zheng (a Chinese sailor) and Yang (a Mongolian sailor) want to explore a rectangular section of the Indian Ocean. The area can be split up into a grid of $N \times M$ square regions with N rows and M columns. Since the Mongols have a hostile relationship with the Ming empire, the two explorers decide to take different paths. Zheng starts from the bottom-right corner and moves horizontally, switching directions when he reaches an edge. Yang starts from the top-left corner and moves vertically, switching directions when he reaches an edge.

Diagrams of the two paths are shown below for the case when $N = 3$ and $M = 5$:



Zheng Yang

Zheng labels the regions on his map from 1 to $N \cdot M$ in the order that he visits them. Yang does the same for his own map. Can you figure out which regions have the same label on both explorers' maps?

Input Format:

The only line contains two integers N and M ($1 \leq N, M \leq 1000$), the number of regions along the height and width, respectively, of the rectangle.

Output Format:

On the first line print the number of regions that have the same label on both Zheng's map and Yang's map.

On the second line print the labels of these regions **in ascending order**.

Subtasks:

1. $M, N \leq 20$ (25% of system tests)
2. No additional constraints (75% of system tests)

Sample Input:

```
3 5
```

Sample Output:

```
3
4 8 12
```

We will use a coordinate system with $(1, 1)$ and $(3, 5)$ as the top-left and bottom-right corners, respectively. Then region at $(3, 2)$ has the label 4 on both maps, region at $(2, 3)$ has the label 8 on both maps, and region at $(1, 4)$ has the label 12 on both maps; these are highlighted in yellow in the example.

§4 Banquet

Marco Polo praised the Chinese city of Kinsay in his travelogue, claiming that it had grand island palaces for marriage feasts and other large events.

Albert works on one of Kinsay's islands. There are N plates evenly spaced out in a line. He needs to reorganize them into stacks of sizes $A_1, \ldots, A_N$ for an upcoming banquet (there must be A_i plates at position i). In one second, Albert can move a single plate from one position to a **neighboring** position. Compute the minimum amount of time needed to organize the plates as desired. All plates are indistinguishable.

Input Format:

The first line contains N ($1 \leq N \leq 10^5$), the total number of plates.

The second line contains N integers $A_1, \ldots, A_N$ ($0 \leq A_i \leq N$), meaning that there must be A_i plates at each position i after Albert is done. It is guaranteed that $\sum_{i=1}^{N} A_i = N$.

Output Format:

Print the number of seconds needed to reorganize the plates if Albert works optimally.

Subtasks:

1. $N \leq 1000$ (25% of system tests)
2. No additional constraints (75% of system tests)

Sample Input:

```
6
0 3 0 0 2 1
```

Sample Output:

```
3
```

If Albert moves the plates at positions 1 and 3 to position 2 using 1 second each and then moves the plate at position 4 to position 5 using another second, he will finish the reorganization in 3 seconds. 3 seconds is the minimum time required.

§5 Climbing Trees

Scientists believe that the last common ancestor of humans and modern chimpanzees lived around 10 million years ago.

Joe the monkey wants to get some exercise by climbing trees in the forest. The forest has N trees with heights $A_1, \ldots, A_N$ meters. Joe is rather unfit, so he can only climb a tree if its height is within K meters (inclusive) of the height of the previous tree he climbed. He also refuses to visit any tree more than once. More formally, Joe will climb a sequence of trees $T_1, \ldots, T_t$ ($1 \le T_i \le N$) such that $|A_{T_{i+1}} - A_{T_i}| \le K$ for $i = 1, \ldots, t-1$ and $T_i \ne T_j$ for $i \ne j$. What is the maximum sum $\sum_{i=1}^{t} A_{T_i}$ of the heights of the trees that Joe climbs? He may start by climbing any tree.

Input Format:

The first line contains N and K ($1 \le N \le 10^5$; $1 \le K \le 10^9$), the number of trees in the forest and the permitted height difference between consecutive trees that Joe climbs.

The next line contains N integers $A_1, \ldots, A_N$ ($1 \le A_i \le 10^9$) representing each tree's height.

Output Format:

Print a single integer representing the maximum total height that Joe can climb.

Sample Input:

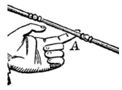
```
5 5
12 18 2 7 4
```

Sample Output:

```
25
```

Joe can start by climbing the 4-meter tree, then the 2-meter tree, then the 7-meter tree, and finally the 12-meter tree for a total height of 25 meters. This is the optimal total height. Note that there is more than one order in which Joe can climb these trees.

§6 Weights



The Greek mathematician and scientist Archimedes of Syracuse (c. 287–212 BCE) is credited with introducing the concept of the center of gravity.

Archimedes has N identical iron weights arranged on a number line. The weights are at the positions $A_1, A_2, \ldots, A_N$. The *center of mass* of a collection of weights is the average of their positions. For example, if there are weights at positions 3, 4, 7, and 12, their center of mass would be 6.5.

Archimedes plans to paint each weight either red or blue in such a way that the center of mass of the red weights and the center of mass of the blue weights are as far apart as possible. There must be at least one weight of each color. In other words, he wants to divide the array $A_1, A_2, \ldots, A_N$ into two non-empty groups such that the difference between the averages of the groups is maximized. What is this largest possible difference? Your output will be accepted if the absolute or relative error is less than 10^{-6}.

Input Format:

The first line contains N, the number of weights ($2 \le N \le 10^5$).

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($-10^9 \le A_i \le 10^9$) specifying the position (x-coordinate) of each weight.

Output Format:

Print the maximum possible distance between the two centers of mass.

Subtasks:

1. $N \le 1000$ (25% of system tests)
2. No additional constraints (75% of system tests)

Sample Input:

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

Sample Output:

```
7.4
```

The maximum distance can be achieved when -6 is painted red while the rest of the weights are painted blue. The centers of mass are simply the averages of the positions: $\frac{-6}{1} = -6$ for red and $\frac{5+0+1+3-2}{5} = 1.4$ for blue. The answer is $1.4 - (-6) = 7.4$, which can be proven to be largest possible difference.

§7 Stone Piles

The stone circle in Nabta Playa (a desert site in southern Egypt) actually predates the Stonehenge by at least two millennia.

Gabe is building a replica of the Stonehenge, but he must first organize the stones properly. There are currently N stones arranged into M piles. In each pile, the stones are stacked on top of each other so that there is always one stone on top (unless the pile is empty, in which case there is no top stone). Each stone has a certain *type*, described by an integer between 1 and M, inclusive. In one action, Gabe can move the top stone from any non-empty pile to the top of another pile. He wants to find a sequence of actions that will result in all stones ending up in their proper stack. That is, pile t must contain all the stones of type t for $t = 1, 2, \ldots, M$. Can you help him find such a sequence? **Your solution does not have to be the shortest possible, but it cannot include more than $5 \cdot 10^5$ actions.**

Input Format:

The first line contains two integers N and M ($1 \le N \le 10^5; 3 \le M \le 10^5$) representing the number of stones and the number of piles.

The next M lines each contain an integer R_i representing the number of stones in the i-th pile, followed by R_i integers $T_{i,1}, \ldots, T_{i,R_i}$ describing the types of the stones in the pile from **top to bottom** ($0 \le R_i \le N$; $1 \le T_{i,j} \le M$). It is guaranteed that there will be N stones in total among all the stacks.

Output Format:

On the first line print K, the number of actions in your solution ($0 \le K \le 5 \cdot 10^5$).

The next K lines should each contain two integers a_i and b_i ($1 \le a_i, b_i \le M$), meaning that as his i-th action Gabe should move the stone on the top of pile a_i to the top of pile b_i.

It can be proven that a valid solution always exists under these conditions.

Subtasks:

1. $N, M \le 1000$ (25% of system tests)
2. No additional constraints (75% of the system tests)

Sample Input:

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

Sample Output:

```
4
2 3
1 3
1 2
3 1
```

The initial configuration of the piles is as follows (an underscore indicates an empty pile):

```
1
2 3 _
```

The following sequence of 4 actions places each stone into its correct pile as desired:

- $2 \longrightarrow 3$
  ```
  1
  2 _ 3
  ```
- $1 \longrightarrow 3$
  ```
      1
  2 _ 3
  ```
- $1 \longrightarrow 2$
  ```
      1
  _ 2 3
  ```
- $3 \longrightarrow 1$
  ```
  1 2 3
  ```

Although this solution can be shown to be the shortest possible, any valid sequence with less than or equal to $5 \cdot 10^5$ actions will be accepted.

§8 Number Game

When German mathematician Carl Friedrich Gauss (1777–1855) was seven, he amazed his teacher by quickly computing the sum $1 + 2 + \cdots + 100$.

One day Gauss's teacher, Mr. Büttner, decides to test Gauss's mathematical skill. First, Mr. Büttner comes up with some nonnegative integer X which he keeps secret. He then goes through each digit of X from left to right: for each digit d, he writes either d^2 or d^3 on the chalkboard. He doesn't leave any space between the numbers he writes, so the resulting values are concatenated into a single giant integer Y. Mr. Büttner then challenges Gauss to figure out X by only knowing Y. Can you help Gauss find **any** possible value of the original number X?

Input Format:

The only line contains the integer Y $(0 \leq Y \leq 10^{(10^5)})$.

Output Format:

Print **any** possible value of X. **If no valid** X **exists, print** `-1`.

Subtasks:

1. $Y \leq 10^{(10^3)}$ (25% of system tests)
2. No additional constraints (75% of system tests)

Sample Input:

```
49216
```

Sample Output:

```
236
```

`236` is a possible value for X because $2^2 = 4$, $3^2 = 9$, and $6^3 = 216$, so concatenating these three results gives `49216`. `76` is also an acceptable answer: $7^2 = 49$ and $6^3 = 216$. These are the only two correct answers for this case.

§9 Heating Rocks

The Greeks enjoyed heated "laconica" bath houses named after the Spartan kingdom of Laconica.

A public bathhouse in ancient Greece uses hot rocks to warm the bath water. Currently, there are N rocks that have the temperatures $T_1, ..., T_N$ (all temperatures are in °C). To get the water to the appropriate warmth, Aaron wants each rock be at least X °C.

To heat the rocks, there is a fireplace that can hold **up to two** rocks at a time. Rocks in the fire get hotter at a rate of 1 C° per second. If Aaron uses the fireplace optimally, what is the minimum amount of time required until all rocks have a temperature of at least X °C? You may assume that the rocks can be put in and taken out of the fireplace instantaneously, although **they may only be switched at integer times**. The temperature of a rock remains constant when it is not in the fire. You may ignore the fact that water turns to plasma at around 12000 °C: this bathhouse is using magical Greek water that can reach any temperature.

Input Format:

The first line contains two integers N and X ($1 \le N \le 10^5; 1 \le X \le 10^9$), the desired temperature of the rocks in °C.

The second line contains N integers: $T_1, ..., T_N$ ($1 \le T_i \le 10^9$), the starting temperatures of the rocks in °C.

Output Format:

Output a single integer, the minimum number of seconds Aaron needs to heat all N rocks to X °C.

Subtasks:

1. $N \le 5$; $X \le 100$ (50% of system tests)
2. No additional constraints (50% of system tests)

Sample Input:

```
5 10
4 6 8 10 10
```

Sample Output:

```
6
```

If Aaron heats the first two stones for 4 seconds, the first will be heated to 8 °C and the second to 10 °C. Aaron can then heat the first and third rocks for 2 seconds, bringing them both to 10 °C. This takes 6 seconds in total, which is the minimum time needed.

§10 Calendars

The Aztecs and the Mayans both used a 365-day solar calendar cycle and a 260-day ritual calendar.

Clarence the historian has become interested in the calendars of ancient civilizations. During his research, he found two mysterious calendars A and B belonging to early cultures. Both calendars have N-day weeks, but the days may appear in different orders in the two calendars. Formally, each calendar can be represented as a permutation of the numbers $1, \ldots, N$.

He has defined some vocabulary to describe calendars for his research. Given any permutation P, let $pos_P(k)$ be the index of k in P (indexing starts at 1). For example, if $P = [2, 4, 1, 5, 3]$ then $pos_P(4) = 2$ because 4 is the second element of P. Clarence then defines the *permutational distance* between any two permutations P and Q (both of size N) to be

$$dist(P, Q) := \sum_{k=1}^{N} |pos_P(k) - pos_Q(k)|.$$

Clarence wants to know how similar his two mysterious calendars are. Unfortunately, the second calendar B is written on a circular tablet, so he doesn't where it starts and ends. Thus, Clarence wants to know the smallest permutational distance that any rotation of B could have with A. That is, he wants to calculate the minimum value of $dist(A, C)$ for any permutation C that is a rotation of B (meaning C can be created from B by successively moving B's first element to the back). Can you help him find this value? Note that A is fixed; Clarence only wants to rotate B.

Input Format:

The first line contains the integer N ($1 \leq N \leq 10^5$).

The second line contains N distinct integers $A_1, \ldots, A_N$ ($1 \leq A_i \leq N$) describing the first calendar cycle.

The third line contains N distinct integers $B_1, \ldots, B_N$ ($1 \leq B_i \leq N$) describing the second calendar cycle.

Output Format:

Print the minimum permutational distance between A and any rotation of B.

Subtasks:

1. $N \leq 1000$ (25% of system tests)
2. No additional constraints (75% of system tests)

Sample Input:

```
4
1 4 2 3
2 4 3 1
```

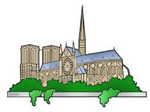
Sample Output:

```
2
```

The permutation `1 2 4 3` is the best choice for C. The permutational distance between `1 4 2 3` and `1 2 4 3` is 2, which is the minimum over all rotations of B.

Note that a permutation is considered a 0-rotation of itself, so C is allowed to be equal to B.

§11 Cathedral



The first Gothic cathedrals in 12th-century France occasionally collapsed because their architects had little formal training in engineering.

Aaron has been tasked with overseeing the construction of arches for a Gothic cathedral in France. The raw material for the arches comes in a row of N rocks with sizes given by $P_1, \ldots, P_N$. At this particular quarry **the rocks slope downward**, so $P_i \geq P_j$ for any $i < j$. Aaron wants to plan the construction of M potential arches which could be added to the cathedral. The i-th arch must be built from the subarray of P starting at index s_i with length $2k_i$: namely $P_{s_i}, P_{s_i+1}, \ldots, P_{s_i+2k_i-1}$.

To build an arch, the first k_i rocks of its subarray will be merged into one pillar and the remaining k_i rocks will be merged into another. The *unevenness* of an arch is the absolute difference between the sizes of its two pillars, given by

$$\left| \sum_{j=s_i}^{s_i+k_i-1} P_j - \sum_{j=s_i+k_i}^{s_i+2k_i-1} P_j \right|.$$

Aaron wants the unevenness of each arch to be as small as possible so that the cathedral remains stable. To achieve this, she can tell her workers to perform some amount of *circulations* on the subarray of an arch. A circulation consists of temporarily removing rock s_i from the row, sliding rocks $s_i+1, \ldots, s_i+2k_i-1$ left by one space, then placing the original rock into position $s_i + 2k_i - 1$ to fill the gap. By performing these circulations, Aaron can effectively achieve any desired rotation of $P_{s_i}, P_{s_i+1}, \ldots, P_{s_i+2k_i-1}$. Note that **circulations only affect rocks in the arch's subarray**, ignoring the other $N - 2k_i$ rocks.

For each arch, tell Aaron the minimum unevenness of $P_{s_i}, P_{s_i+1}, \ldots, P_{s_i+2k_i-1}$ she can obtain by applying any amount of circulations (possibly zero) to the subarray. Since Aaron is only planning out potential arches, **the entire array P resets after each arch is processed**. Thus, each arch can be treated as an independent query.

Input Format:

The first line contains the two integers N and M $(1 \leq N, M \leq 10^5)$.

The second line contains N integers $P_1, \ldots, P_N$ $(0 \leq P_i \leq 10^9)$ representing the sizes of the rocks. These integers are **nonincreasing**.

The next M lines each contain two integers s_i and k_i $(1 \leq s_i < s_i + 2k_i - 1 \leq N)$ representing the starting index and half of the length of the subarray.

Output Format:

Print M lines, where the i-th line contains the minimum unevenness of the i-th subarray after Aaron applies some number of circulations to it.

Subtasks:

1. $N, M \leq 1000$ (25% of system tests)
2. No additional constraints (75% of system tests)

Sample Input:

```
8 3
12 9 8 8 7 4 4 3
2 2
3 3
1 2
```

Sample Output:

```
0
4
3
```

For the first arch, we consider the subarray with starting index 2 and length 4 (`9 8 8 7`). Its current unevenness is $|(9+8)-(8+7)| = 2$, but if we circulate it, it becomes `7 9 8 8`, which has an unevenness of $|(7+9)-(8+8)| = 0$.

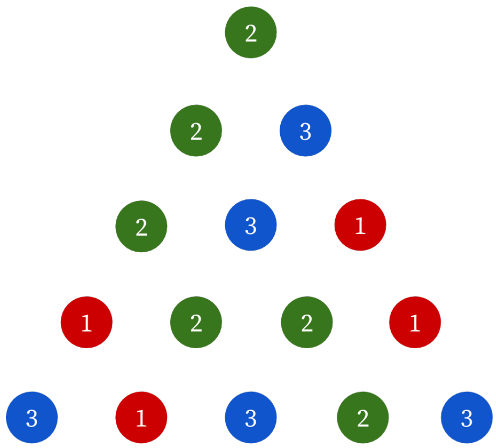
After the circulation from the first arch, the whole array resets. The second arch uses the subarray `8 8 7 4 4 3`; it can be proven that the minimum unevenness is 4, first obtained after 1 circulation.

The third arch uses the subarray `12 9 8 8`, which has a minimum unevenness of 3 after one circulation.

§12 Gemstones

Under the Delhi Sultanate, the Gujarati city of Cambay exported vast quantities of luxury products, including polished gemstones, pearls, and silk cloth.

Maxwell hears of the wonderful business opportunities in Cambay and resolves to pursue a career as a gemstone merchant. Owing to his idiosyncrasies, he has opened a gem mine in the shape of an equilateral triangle with N gems on each side. Each gem comes in one of three colors. An example array with $N = 5$ is shown below:



Maxwell wants to use pieces of string to connect the gems into bracelets, but he has some important criteria (again owing to his idiosyncrasies):

- Each string must be a straight line segment passing through **at least two gems**.
- Each string must start on a gem and end on a gem.
- Each string must be parallel to one side of the triangular mine.
- No two strings may intersect, including at endpoints (no gem may touch more than one string).
- For variety, each string must have a **different** length.
- The total number of strings must be maximized.

After Maxwell is done, he turns each piece of string into a bracelet containing all of the gems it connects. Each bracelet is assigned a price equal to *the number of times the most common color of the bracelet occurs*. In other words, if a string connects r red gems, b blue gems, and g green gems, the corresponding bracelet has a price of $\max(r, g, b)$. Find the maximum sum of bracelet prices Maxwell can obtain under these conditions.

Input Format:

The first line contains the integer N $(1 \leq N \leq 400)$.

Among the next N lines, the i-th line contains i integers $a_{i,1}, \ldots, a_{i,i}$ $(1 \leq a_{i,j} \leq 3)$ specifying (from left to right) the color of the gems in the i-th row from the top.

Output Format:

Print the maximum total price of the bracelets that Maxwell can obtain.

Subtasks:

1. $N \leq 80$ (25% of system tests)
2. $N \leq 200$ (25% of system tests)
3. No additional constraints (50% of system tests)

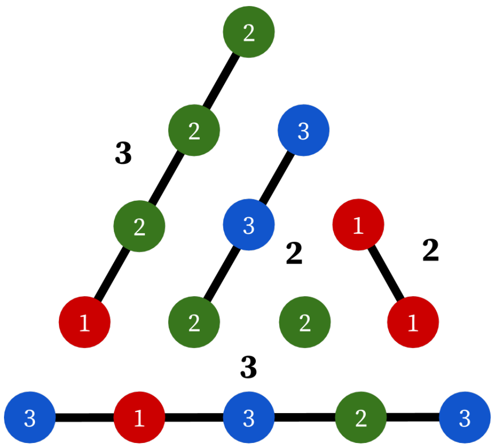
Sample Input:

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

Sample Output:

```
10
```

This array is the same as that in the problem description. It can be shown that the maximum total price is 10, which can be achieved when the gems are strung as shown:



The number next to each string is the price of the bracelet created from that string. It can be shown that this configuration maximizes the total number of strings, which is one of Maxwell's requirements.