Problem A. Hero Wars

Problem Description

Halloween was approaching, and when Monkey finished his shift at the "Animal World Sanctuary" zoo, he strolled through the streets of the Xinyi district. He was surprised to see everyone dressed up as various monsters and creatures, as if he had stepped into a world filled with elaborate costumes. Monkey couldn't help but sigh, once again spending the holiday alone. However, as a hardworking office worker, he was exhausted and didn't have the energy to participate in Halloween activities.

When he got back home, he opened up YouTube. Since he had recently entered the workforce and couldn't afford to purchase YouTube Premium, he had to endure some advertisements. At that moment, he watched an ad. In the video, the protagonist, starting from a very low initial level, challenged monsters under a big demon's attack and gradually leveled up until rescuing the princess. But as you may know, the ad suddenly ended halfway through, and the protagonist didn't even have time to save the princess.

To get into the Halloween spirit, Monkey downloaded the game and started playing. However, after a long day at the office, he gradually grew tired and eventually collapsed halfway through the game. When he opened his eyes again, he found himself in the virtual world of this Halloween game. It turned out he had been reincarnated, and he was the main character. In front of him were numerous monsters, each dressed in Halloween costumes.

Starting from his initial level, Monkey had to face n Halloween monsters, with v_i representing the level of the i^{th} monster. When Monkey's level was greater than or equal to the level of a Halloween monster, he could challenge that monster and absorb its level, causing the monster to disappear permanently in that reincarnation. He can challenge the monsters in **any order**. When Monkey's level reached the target level, he would win the reincarnation world and complete the game.

Monkey went through multiple cycles of reincarnation, with a total of q reincarnations. In each reincarnation, the Halloween monsters were reset to their original levels and quantities. However, Monkey's initial level x_i and target level y_i were different for each reincarnation.

Considering that Monkey was exhausted and wanted to breeze through the game, can you help him determine the minimum number of challenges required to complete each reincarnation? If it's not possible to complete the game, please answer with -1.

Input Format

- line 1: $n \ \ q$
- line 2: $v_1 \ \ v_2 \ \ \ldots \ \ v_n$
- line $2+i$ ($1 \leq i \leq q$): $x_i \ \ y_i$

Output Format

- line i ($1 \leq i \leq q$): the minimum number of challenges required to complete i^{th} reincarnation.

Constraints

- $1 \leq n \leq 100\,000$.
- $1 \leq q \leq 100\,000$.
- $1 \leq v_i \leq 10^9$ for $i = 1, 2, \ldots, n$.
- $1 \leq x_i < y_i \leq 10^{14}$ for $i = 1, 2, \ldots, q$.
- All input values are integers.

Subtasks

1. (15 points) $v_i \leq v_{i+1} \leq 2 \cdot v_i$ for $i = 1, 2, \ldots, n-1$.
2. (30 points) $n \leq 3000$; $q \leq 3000$.
3. (25 points) $x_i \geq 10^9$ for $i = 1, 2, \ldots, q$.
4. (30 points) No additional constraints.

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
Samples	1-3	1000	262144
1	4-12	1000	262144
2	13-21	1000	262144
3	22-27	1000	262144
4	1-40	1000	262144

Samples

Sample Input 1

```
6 2
2 4 8 16 32 64
1 50
9 50
```

This sample input satisfies the constraints of Subtasks 1, 2, 4.

Sample Output 1

```
-1
3
```

Sample Input 2

```
7 6
1 3 3 6 14 18 30
1 3
2 50
3 15
4 60
5 80
10 70
```

This sample input satisfies the constraints of Subtasks 2, 4.

Sample Output 2

```
-1
7
3
5
7
4
```

Sample Input 3

```
4 3
1 1000 1000000 1000000000
9998998998999 100000000000000
9999999999999 100000000000000
9876543210987 100000000000000
```

This sample input satisfies the constraints of Subtasks 2, 3, 4.

Sample Output 3

```
-1
-1
-1
```

Problem B. Intel CPU

Problem Description

When the dark Halloween night descended, Intel Elves and AMD Elves gathered together, intensifying their fierce battle. Under the moonlight, they engaged in a heart-pounding showdown, seamlessly combining the powers of technology and magic.

Intel Elves wielded the enchanted code like ghosts, swift and agile, their eyes shining with infinite wisdom. Meanwhile, AMD Elves controlled mysterious energy, sparking into lightning bolts, showcasing unparalleled speed and strength.

In this dark night sky, their battle depicted the enchanting fusion of technology and magic. Intel Elves' Code Swords clashed with AMD Elves' Thunder Hammers, transforming into magnificent magical displays, stunning the entire Halloween night.

Then the wicked witch KCW cast an evil spell, transforming Intel and AMD Elves into Intel CPUs, helping KCW lab create a joyful candy known as "algorithm." The reason why KCW don't transform to AMD CPUs since he doesn’t like using AMD CPUs.

Instead of using AMD CPUs. KCW loves using Intel CPUs in his lab. Now, in order to make more "algorithm" candies to make the world wonderful, he needs more Intel CPUs.

KCW plans to participate in n auctions. The auctions will be held at very specific times. There are also some additional rules need to follow.

1. You can only join **one auction** at the same time.
2. Once you decided to join an auction, you must attend it from the beginning until the end.
3. The i^{th} auction starts at time s_i and ends at time $e_i + 0.5$.
4. You can earn x_i Intel CPUs after participating in the i^{th} auction.

As a good programmer, can you help KCW find out the maximum number of Intel CPUs he can earn if he participates optimally?

Input Format

- line 1: n
- line $1 + i$ ($1 \leq i \leq n$): s_i e_i x_i

Output Format

- line 1: the maximum number of Intel CPUs KCW can earn.

Constraints

- $1 \leq n \leq 500\,000$.
- $1 \leq s_i \leq e_i \leq 1\,000\,000\,000$ for $i = 1, 2, \ldots, n$.
- $1 \leq x_i \leq 1\,000\,000\,000$ for $i = 1, 2, \ldots, n$.
- All inputs are integers.

Subtasks

1. (40 points) $x_i = 1$ for $i = 1, 2, \ldots, n$.
2. (60 points) No additional constraints.

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
Samples	1-2	1500	262144
1	3-17	1500	262144
2	1-32	1500	262144

Samples

Sample Input 1

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

This sample input satisfies the constraints of all the subtasks.

Sample Output 1

```
2
```

KCW can join 1^{st} and 3^{rd} auction and get 2 Intel CPUs.

Sample Input 2

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

This sample input satisfies the constraints of Subtask 2.

Sample Output 2

```
8
```

KCW can join 2^{nd} and 4^{th} auction and get 8 Intel CPUs.

Problem C. Pecola or Treat

Problem Description

On a serene Halloween night, bathed in the gentle glow of the moonlight, Pecola arrived at the hallo office, quietly prepared to test her newly developed Usada Virus. This virus possessed a unique capability - it could transform infected individuals into adorable rabbit-like creatures. Pecola's intention was to inject some fun and surprise into the upcoming Halloween party.

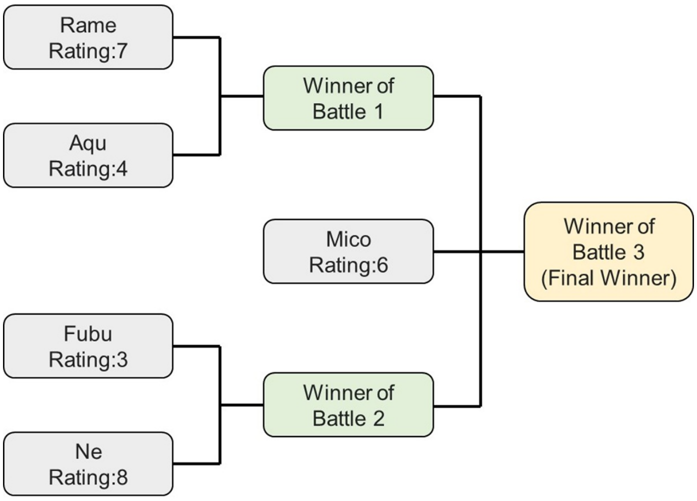
However, as she carefully opened the container containing the virus, a small mishap caused it to slip from her hand, releasing the virus in a mist that swiftly spread throughout the entire office. Before long, every corner of the hallo office was filled with a brief mist, and everyone present had been infected by the magical Usada Virus, transforming into cute rabbit-like creatures.

Pecola found the transformation of everyone quite amusing and let out her signature arrow-like laughter, "HA↑HA↓HA↑HA↓HA↑." However, the abundance of rabbit transformations began to diminish the uniqueness of being a rabbit representative. Consequently, she decided to keep the rabbit form for only one individual and forcibly inject antibodies into the rest. In her role as the internet's giant rabbit, she promptly conceived a new project called "Pecola or Treat" to select members who would retain their rabbit form, serving as compensation for the virus spreading.

The project's specifics were as follows:

- Members would engage in a multiplayer war, and the last person standing would keep their rabbit form.
- Due to the limited size of the arena, **at most** k individuals could participate in each battle.
- In each battle, only one person would emerge as the winner, while the remaining $k-1$ individuals would be forcibly injected with the antibody, losing their rabbit form. The battles would continue repeatedly until only one member retained their rabbit form as the ultimate winner.
- As a participation reward and compensation for being infected by the virus, Pecola would bestow upon each individual $a_i \cdot \text{dis}(i)$ diamonds, where a_i represented each person's rabbitization rating, and $\text{dis}(i)$ indicated the maximum number of battles they needed to participate in.

For instance, when there are 5 members in an arena that can accommodate 3 participants for the battle, here is a possible schedule:



In Rame's case, her Rabbitization rating is 7, and if she advances successfully each time, she would participate in a maximum of 2 battles. Therefore, Pecola would pay her $7 \times 2 = 14$ diamonds.

Following this pattern, Pecola would pay a total of $(7+4+3+8) \times 2 + 6 \times 1 = 50$ diamonds.

With numerous victims, Pecola would have to mine on Neko Neko Island relentlessly to repay them. Although Pecola couldn't change the rabbitization rating of the members, as the organizer, she had the power to determine the schedule. Please assist Pecola in designing a schedule that minimizes the number of diamonds she needs to pay, thereby averting the unfortunate fate of a lifetime of mining on Neko Neko Island.

Input Format

- line 1: $n \ k$
- line 2: $a_1 \ a_2 \ \ldots \ a_n$

Output Format

- line 1: ans : the minimum number of diamonds Pecola needs to pay
- line 2: m : number of battles in your answer
- line $2+i$ ($1 \le i \le m$): $k_i \ p_{i,1} \ p_{i,2} \ \ldots \ p_{i,k_i}$: number of players in the i^{th} battle, followed by the list of players
  - Please use $n+i$ to represent the winner of the i^{th} battle.

Your output must satisfy:

- Numbers in $[1, n+m-1]$ must appear exactly once in $p_{i,j}$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, k_i$.
- $2 \le k_i \le n$.
- $1 \le p_{i,j} \le n+i-1$ for $j = 1, 2, \ldots, k_i$.

Constraints

- $1 \le n \le 1\,000\,000$.
- $2 \le k \le 1\,000\,000$.
- $0 \le a_i \le 1\,000\,000\,000$ for $i = 1, 2, \ldots, n$.
- All inputs are integers.

Subtasks

1. (5 points) $n \le k$.
2. (35 points) $k = 2$.
3. (15 points) $n \equiv 1 \bmod (k-1)$.
4. (45 points) No additional constraints.

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
Samples	1-3	2000	262144
1	4-9	2000	262144
2	10-15	2000	262144
3	16-21	2000	262144
4	22-34	2000	262144

Samples

Sample Input 1

```
5 3
4 8 7 6 3
```

This sample input satisfies the constraints of Subtasks 3, 4.

Sample Output 1

```
41
2
3 5 1 4
3 6 3 2
```

Sample Input 2

```
6 3
1 1 1 11 1 1
```

This sample input satisfies the constraints of Subtask 4.

Sample Output 2

```
21
3
3 1 2 3
2 5 6
3 7 4 8
```

Sample Input 3

```
2 2
5 2
```

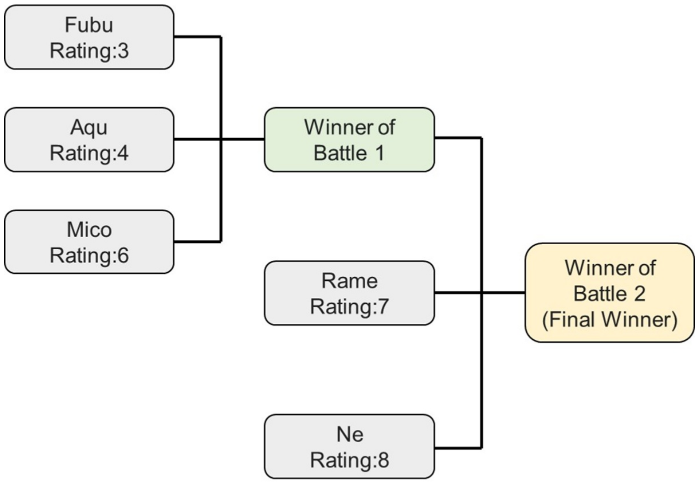
This sample input satisfies the constraints of all the subtasks.

Sample Output 3

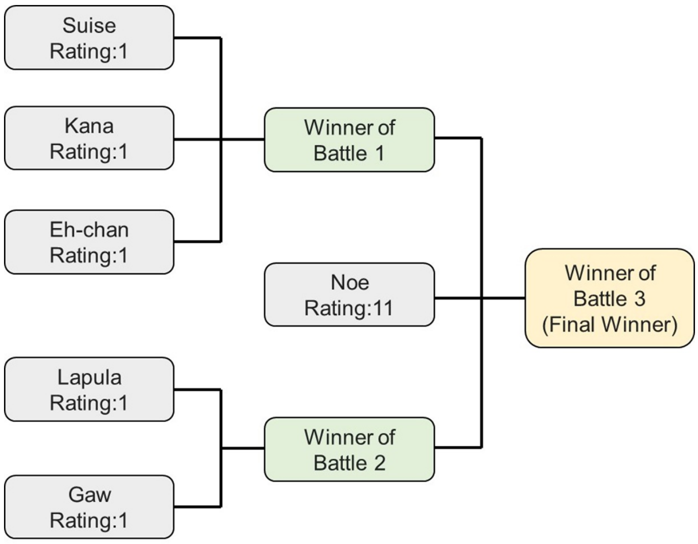
```
7
1
2 2 1
```

Notes

One of the best schedule of Sample 1:



One of the best schedule of Sample 2:



Problem D. Thrill Factor

Problem Description

As Halloween approaches in a quaint, peaceful town, children eagerly prepare for a night of door-to-door trick-or-treating, infusing the town with both fear and excitement. As the orchestrator of this special event, your mission is to ensure that this night brims with spine-tingling scares.

Within the town, you'll find m children, each possessing varying levels of "trick-or-treat potential." Additionally, there are n households awaiting their share of Halloween mischief, each brimming with fear.

You have thoughtfully assembled a collection of $n \times m$ distinctive Halloween costumes for this event. Prior to the festivities, you will assign a unique number to each child, ensuring they queue up to receive their costumes in an order of your choosing. This "trick-or-treat potential" can be represented as a vector v with dimensions $m \times 1$, and you have the flexibility to arrange this order to your liking.

The event encompasses a total of n rounds of trick-or-treating. During each round, children will form a queue to obtain their Halloween costumes, creating a matrix A with dimensions $n \times m$. Here, $A_{i,j}$ signifies the "scariness" of the costume donned by the j-th child during the i-th round of trick-or-treating. The order in which the costumes are used is at your discretion. To keep the event fresh, each costume is designated for a single use, ensuring that each Halloween costume is employed exactly once and not repeated.

The "spookiness" of each round of trick-or-treating can be quantified. After the children have prepared for each round, you can compute the "spookiness" by multiplying the "trick-or-treat potential" of each child by the "scariness" of their costume and then summing these values. The result is a vector of dimensions $n \times 1$, referred to as Av, which sequentially represents the "spookiness" of each round.

Since each household has a distinct fear level when it comes to trick-or-treating, the same round of pranks may have different effects on different households. This variation can be quantified as the "thrill factor." For any given round of trick-or-treating, the "thrill factor" is determined by multiplying the household's "fear level" by the "spookiness" of that round. The order in which you visit the n households is entirely your decision, assuming that, according to your arrangement, the "fear level" of each household can be represented as a vector with dimensions $n \times 1$, denoted as u. Consequently, the overall "thrill factor" for the entire event is represented as $u^{\mathbf{T}}(Av)$.

Your goal is to obtain the maximum value of the overall "thrill factor" $u^{\mathbf{T}}Av$ for the entire event. You are to provide the "trick-or-treat potential" v in the order in which the children should queue up, the "scariness" A of the costumes based on the order they are worn, and the "fear level" u of each household, based on your chosen order for trick-or-treating. If multiple arrangements lead to the maximum value, you have the flexibility to output any one of them.

Please keep in mind that your goal is to maximize the "thrill factor" for the entire town without arbitrarily altering the children's "trick-or-treat potential," the scariness of the prepared costumes, or the fear level of each household. Make sure that this Halloween night becomes an unforgettable experience.

Input Format

- Line 1: Two integers, n and m, representing the number of households and children, respectively.
- Line 2: A list of n integers, $u_1, u_2, \ldots, u_n$, denoting the fear levels of each household.
- Line 3: A list of m integers, $v_1, v_2, \ldots, v_m$, indicating the trick-or-treat potential of each child.
- Lines $3 + i$ (where $1 \leq i \leq n$): A list of m integers, $A_{i,1}, A_{i,2}, \ldots, A_{i,m}$, representing the scariness of costumes worn by each child. Please note that you can rearrange the elements in the matrix, changing both rows and columns as needed for optimization.

Output Format

- Line 1: The maximum value of $u^{\mathrm{T}} A v$.

Subsequently, please output the modified u', v', and A' that result in the maximum value.

- Line 2: A list of n integers, $u'_1, u'_2, \ldots, u'_n$.
- Line 3: A list of m integers, $v'_1, v'_2, \ldots, v'_m$.
- Lines $3 + i$ (where $1 \leq i \leq n$): A list of m integers, $A'_{i,1}, A'_{i,2}, \ldots, A'_{i,m}$, representing the scariness of costumes worn by each child during the i-th round of trick-or-treating.

Constraints

- $1 \leq n, m \leq 10$.
- $|u_i| \leq 10,000$ for $i = 1, 2, \ldots, n$.
- $|v_i| \leq 10,000$ for $i = 1, 2, \ldots, m$.
- $|A_{i,j}| \leq 10,000$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.
- All inputs are integers.

Subtasks

- (20 points) You are not required to rearrange any elements in the matrix.
- (10 points) $n = 1$.
- (10 points) $u_i = 1$ for $i = 1, 2, \ldots, n$.
- (10 points) u, v, and A consists of non-negative integers.
- (50 points) No additional constraints.

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
Samples	1-5	1000	262144
1	6-11	1000	262144
2	12-15	1000	262144
3	16-20	1000	262144
4	21-25	1000	262144
5	1-29	1000	262144

Samples

Sample Input 1

```
9 8
0 -7 7 3 -4 7 -3 1 -9
-7 0 0 7 -9 -8 -5 -2
-3 -2 -1 -1 -1 -1 0 0
8 0 0 -8 9 8 7 4
-7 1 1 8 -9 -9 -7 -4
-5 1 1 5 -5 -5 -5 -3
6 2 2 -6 8 7 5 4
-7 2 2 8 -9 -8 -6 -4
5 3 3 -5 6 6 5 3
-3 3 3 3 -4 -3 -3 -3
9 3 3 -9 9 9 8 5
```

This sample input satisfies the constraints of Subtasks 1, 5.

Sample Output 1

```
11454
0 -7 7 3 -4 7 -3 1 -9
-7 0 0 7 -9 -8 -5 -2
-3 -2 -1 -1 -1 -1 0 0
8 0 0 -8 9 8 7 4
-7 1 1 8 -9 -9 -7 -4
-5 1 1 5 -5 -5 -5 -3
6 2 2 -6 8 7 5 4
-7 2 2 8 -9 -8 -6 -4
5 3 3 -5 6 6 5 3
-3 3 3 3 -4 -3 -3 -3
9 3 3 -9 9 9 8 5
```

Sample Input 2

```
1 3
1
-4 -2 -3
6 3 -4
```

This sample input satisfies the constraints of Subtasks 2, 5.

Sample Output 2

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

Sample Input 3

```
3 3
1 1 1
0 -8 9
5 3 -5
2 0 -2
-2 -4 1
```

This sample input satisfies the constraints of Subtasks 3, 5.

Sample Output 3

```
178
1 1 1
0 9 -8
-2 2 -5
0 3 -4
1 5 -2
```

Sample Input 4

```
2 3
1 2
1 7 4
1 2 3
4 5 6
```

This sample input satisfies the constraints of Subtasks 4, 5.

Sample Output 4

```
169
2 1
1 4 7
2 5 6
1 3 4
```

Sample Input 5

```
2 2
-4 -5
-3 -6
-1 -9
-9 -3
```

This sample input satisfies the constraints of Subtask 5.

Sample Output 5

```
-345
-5 -4
-3 -6
-9 -1
-9 -3
```

Problem E. Path of the Wealth

- 2023.11.04 08:45 Update: Added Sample 3.
- 2023.11.04 08:45 Update: Updated problem statement.

Problem Description

Intrigued by the famous (or maybe notorious) "*Neko-chan Chronicle*," you couldn't wait to know what happened after the HallowLive squad beat the final trial. Therefore, you dig into its next chapter, Chapter 76, the "Path of the Wealth."

In the "*Path of the Wealth*," the squad members faced with another puzzle consisting of several mazes, where the treasures are distributed in several rooms. After exploration, Neko-chan finds a sign at the entrance, which reads:

Brave warriors came along,

To claim treasures sleep for a long while.

Marine, the greatest pirate, stand in their way,

Waiting for her next innocent prey.

With ancient artifacts closely held,

They faced their fears and finally prevailed.

Remember my word, brave warriors,

Infinite wealth represents the ultimate power.

The knowledgeable scholar, Peko-chan, translate the ancient poem into modern language:

- There are several rooms in each maze, placed with gold or artifact.
- To defeat ***Marine, the greatest pirate,*** they have to claim as much gold to smash the gold on her face.
- There are also rooms placed with an artifact, which will multiply the wealth of the warriors by specific times.
- You must claim the gold/artifact once you entered the room, and the room will not be reset.

Careful Laffey also finds that there are t mazes in total, each has n rooms. In each maze, room i has exactly one prerequisite room p_i which should be visited before room i. Only room 1 has $p_1 = 0$, indicating that it can be visited directly in the beginning.

Initially, the HallowLive squad has only 1 gold. Can you help Neko-chan calculate the maximum amount of gold they can take away in the maze and the right order to go through each maze?

Input Format

- line 1: t

There will be t blocks following by, each describing a maze. The format of the block is as below:

- line 1: n

n denotes the number of rooms in the maze.

- line 2: $p_1 \ p_2 \ \ldots \ p_n$

Each denote the index of the previous room connected to the current one.

- line $2+i$ ($1 \le i \le n$): $c_i \ v_i$

c_i denotes whether the room has a treasure or an artifact, v_i denotes the value it contains.

Output Format

Print t blocks, one for each testcase. The format of the block is as below:

- line 1: the maximum gold the HallowLive squad can earn. If it is at least 10^{18}, output **`infinite`**.
- line 2: $\ell_1 \ \ell_2 \ \ldots \ \ell_n$: the orders in your plan. If the first line is **`infinite`**, you should output any valid plan that can earn at least 10^{18} gold.

Constraints

- $1 \le t \le 40$.
- $3 \le n \le 5000$.
- $p_1 = 0$.
- $1 \le p_i < i$ for $i = 2, 3, \ldots, n$.
- $c_i \in \{\texttt{+}, \texttt{x}\}$ for $i = 1, 2, \ldots, n$.
- $1 \le v_i \le 9$ for $i = 1, 2, \ldots, n$.
- All input values except c_i ($1 \le i \le n$) are integers.

Subtasks

1. (10 points) $n \le 8$; the maximum answer is less than 10^{18}.
2. (30 points) $n \le 300$; the maximum answer is less than 10^{18}.
3. (10 points) $p_i = 1$ for $i = 2, 3, \ldots, n$.
4. (20 points) n is odd; $p_i = 1$ and $c_i = \texttt{x}$ for $i = 2, 4, \ldots, n-1$; $p_i = i - 1$ and $c_i = \texttt{+}$ for $i = 3, 5, \ldots, n$.
5. (20 points) $|\{k : p_k = i\}| \le 1$ for $i = 2, 3, \ldots, n$.
6. (10 points) No additional constraints.

If you find out the maximum gold the HallowLive squad can earn correctly, but failed to construct a valid plan that yields that amount of gold, you can still get 60% points by printing any valid plan (e.g. `1 2 3 ...` n).

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
Samples	1-3	1000	262144
1	4-12	1000	262144
2	4-20	1000	262144
3	21-24	1000	262144
4	25-28	1000	262144
5	21-35	1000	262144
6	1-40	1000	262144

Samples

Sample Input 1

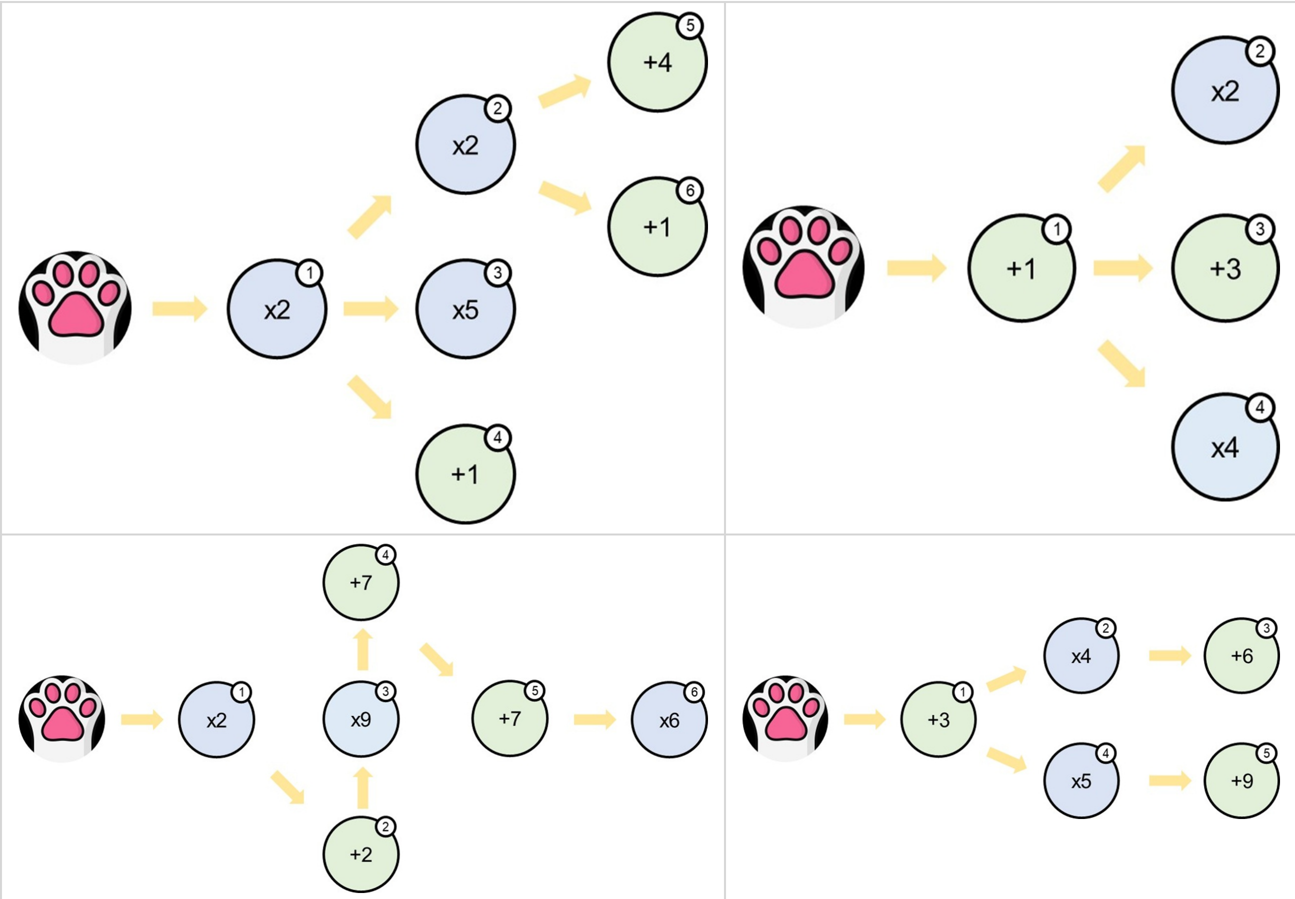
```
4
6
0 1 1 1 2 2
x 2
x 2
x 5
+ 1
+ 4
+ 1
4
0 1 1 1
+ 1
x 2
+ 3
x 4
6
0 1 2 3 4 5
x 2
+ 2
x 9
+ 7
+ 7
x 6
5
0 1 2 1 4
+ 3
x 4
+ 6
x 5
+ 9
```

This sample input satisfies the constraints of Subtasks 1, 2, 6.

Sample Output 1

```
55
1 4 2 5 6 3
40
1 3 2 4
300
1 2 3 4 5 6
122
1 4 5 2 3
```

The four testcases in Sample Input 1 are illustrated below:



Sample Input 2

```
1
19
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
x 9
```

This sample input satisfies the constraints of Subtasks 5, 6.

Sample Output 2

```
infinite
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

Sample Input 3

```
1
5
0 1 1 2 3
x 1
+ 1
+ 1
x 2
x 2
```

This sample input satisfies the constraints of Subtasks 1, 2, 5, 6.

Sample Output 3

```
12
1 2 3 4 5
```