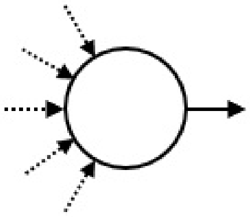
Механикалык оюнчук

нүө ӨҮҢ Жылуулардын атайын удаалаштыгын автоматча аткарууучу оюнчук механикалык оюнчук деп аталат. Жапанда, байрыкыдан көп механикалык оюнчук жасалды.

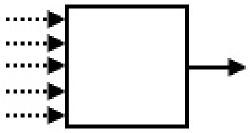
Механикалык оюнчуктун жылуулары **түзүлүштөрдөн** турган **Агым** аркылуу башкарылат. Түзүлүштөр трубкалар аркылуу байланган. Ар бир түзүлүш бир канча (нөл болуш мүмкүн) **Кириш**ке жана бир же эки **Чыгыш**ка ээ. Ар бир трубка түзүлүштүн Чыгышын ошол эле же башка түзүлүштүн Кириши менен байлайт. Ар бир Кириш жана ар бир Чыгыш менен бир гана трубка байланат.

Оюнчук жылууларын кантип аткарууну жазуу үчүн, түзүлүштөрдүн биринде коюлган **шар**ды элесте. Шардын саякатынын ар бир кадамда, шар түзүлүштөн анын Чыгыштарынын менен чыгып, ал Чыгышка байланган трубка боюнча барып, трубканын башка аягына байланган түзүлүшкө кирет.

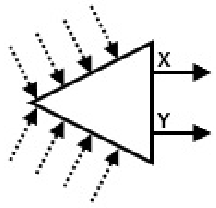
Түзүлүштөрдүн **Баштоо**, **Жандыруу** жана **Алмашуу** үч түрү бар. Бир гана Баштоо, M Жандыруу жана S Алмашуу (S нөл мүмкүн). Сен Sтин маанисин чечиш керек. Ар бир түзүлүш сейрек номурга ээ. Адегенде шар коюлган түзүлүш Баштоосу бар. Ал бир гана Чыгышка ээ. Анын номуру 0.



Шар Жандырууга киргенден кийин жылуунун бир түрүн аткарыш керек. Ар бир түзүлүш бир гана Чыгышка ээ. Жандыруулардын номурлары 1ден Mге дейре.



Ар бир алмаштыруу 'X' жана 'Y' аталган эки Чыгышка ээ. Алмашуунун **Болушу** 'X' же 'Y'. Шар Алмашууга кирип, Алмашуудан азыркы Болуш аныкталган Чыгыш аркылуу чыгат. Андан кийин, Алмашуу өзүнүн Болушун карама-каршыга алмаштырат. Адегенде, ар бир Алмашуунун Болушу 'X'. Алмашуулардын номурлары (-1)ден $(-S)$ке дейре.



Сага Жандыруулардын M саны берилет. Дагы, сага узундугу N болгон A удаалаштыгы берилет. Анын ар бир мүчөсү Жандыруунун номуру. Ар бир Жандыруу A удаалаштыгында бир канча (нөл мүмкүн) жолу бар. Сенин тапшырмаң төмөнкү шарттарды канааттандыруучу саякатты жасоо:

- Шар бир канча кадамдан кийин Баштоого кайта келет.
- Шар биринчи жолу Баштоого кайта келгенде, ар бир Алмашуунун болушу 'X'.
- Шар Жандырууларга так N жолу кирип, биринчи жолу Баштоого кайта келет. Бул Жандыруулардын *удаа* номурлары $A_0, A_1, \ldots, A_{N-1}$.
- Шар биринчи жолу Баштоого кайта келгенче, Алмашуулардын Болуштарын алмаштырууларынын жалпы саны P болсун. P <= 20 000 000 керек.

Ошондой эле, өтө көп Алмашууну колдонгуң келбейт.

Implementation details

You should implement the following procedure (аты: Агымды-жаса).

```
create_circuit(int M, int[] A)
```

- `M`: the number of triggers (Жандыруулардын саны).
- `A`: an array of length N, giving the consecutive serial numbers of the triggers the ball needs to enter.
- This procedure is called exactly once.
- Note that the value of N is the length of the array `A`, and can be obtained as indicated in the implementation notice.

Your program should call the following procedure to answer.

```
answer(int[] C, int[] X, int[] Y)
```

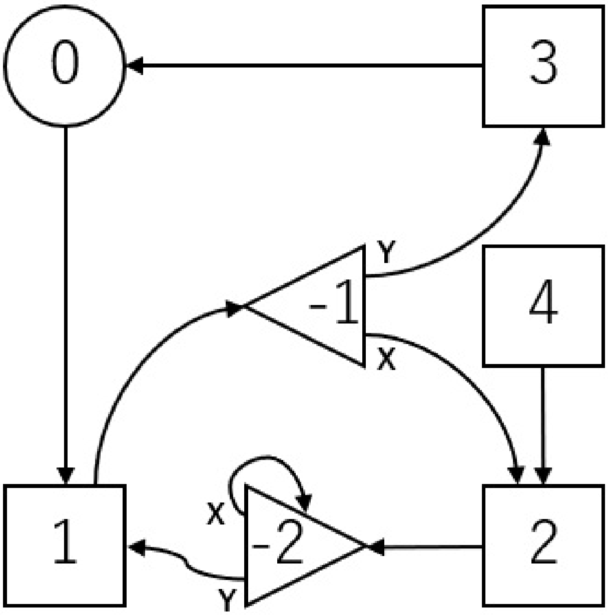
- `C`: an array of length $(M+1)$. The exit of the device (түзүлүштүн Чыгышы) i ($0 \leq i \leq M$) is connected to the device `C[i]`.
- `X`, `Y`: arrays of the same length. The length S of these arrays is the number of the switches. For the switch $-j$ ($1 \leq j \leq S$), its exit 'X' is connected to the device `X[j - 1]` and its exit 'Y' is connected to the device `Y[j - 1]`.
- Every element of `C`, `X`, and `Y` must be an integer between $-S$ and M, inclusive.
- S must be at most 400 000.

- This procedure must be called exactly once.
- The circuit represented by `C`, `X`, and `Y` must satisfy the conditions in the problem statement.

If some of the above conditions are not satisfied, your program is judged as **`Wrong Answer`**. Otherwise, your program is judged as **`Accepted`** and your score is calculated by S (see Subtasks).

Example

Let $M = 4$, $N = 4$, and $A = [1, 2, 1, 3]$. The grader calls `create_circuit(4, [1, 2, 1, 3])`.



The above figure shows a circuit, which is described by a call `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. The numbers in the figure are the serial numbers of the devices.

Two switches are used. Thus $S = 2$.

Initially, the states of the switches -1 and -2 are both 'X'.

The ball travels as follows:

$$0 \longrightarrow 1 \longrightarrow -1 \overset{\text{X}}{\longrightarrow} 2 \longrightarrow -2 \overset{\text{X}}{\longrightarrow} -2 \overset{\text{Y}}{\longrightarrow} 1 \longrightarrow -1 \overset{\text{Y}}{\longrightarrow} 3 \longrightarrow 0$$

- When the ball first enters the switch -1, its state is 'X'. Hence, the ball travels to the trigger 2. Then the state of the switch -1 is changed to 'Y'.
- When the ball enters the switch -1 for the second time, its state is 'Y'. Hence, the ball travels to the trigger 3. Then the state of the switch -1 is changed to 'X'.

The ball first returns to the origin, having entered the triggers $1, 2, 1, 3$. The states of

the switches -1 and -2 are both 'X'. The value of P is 4. Therefore, this circuit satisfies the conditions.

The file `sample-01-in.txt` in the zipped attachment package corresponds to this example. Other sample inputs are also available in the package.

Constraints

- $1 \le M \le 100\,000$
- $1 \le N \le 200\,000$
- $1 \le A_k \le M$ $(0 \le k \le N-1)$

Subtasks

The score and the constraints for each test case are as follows:

1. (2 points) For each i $(1 \le i \le M)$, the integer i appears at most once in the sequence $A_0, A_1, \ldots, A_{N-1}$.
2. (4 points) For each i $(1 \le i \le M)$, the integer i appears at most twice in the sequence $A_0, A_1, \ldots, A_{N-1}$.
3. (10 points) For each i $(1 \le i \le M)$, the integer i appears at most 4 times in the sequence $A_0, A_1, \ldots, A_{N-1}$.
4. (10 points) $N = 16$
5. (18 points) $M = 1$
6. (56 points) No additional constraints

For each test case, if your program is judged as **`Accepted`**, your score is calculated according to the value of S:

- If $S \le N + \log_2 N$, you gain the full score for the test case.
- For each test case in Subtasks 5 and 6, if $N + \log_2 N < S \le 2N$, you gain a partial score. The score for the test case is $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N}\right)^2$, multiplied by the score assigned to the subtask.
- Otherwise, the score is 0.

Note that your score for each subtask is the minimum of the scores for the test cases in the subtask.

Sample grader

The sample grader reads the input from the standard input in the following format.

- line 1: M N
- line 2: A_0 A_1 $\ldots$ A_{N-1}

The sample grader produces three outputs.

First, the sample grader outputs your answer to a file named `out.txt` in the following format.

- line 1: S
- line $2 + i$ ($0 \leq i \leq M$): `C[i]`
- line $2 + M + j$ ($1 \leq j \leq S$): `X[j - 1] Y[j - 1]`

Second, the sample grader simulates the moves of the ball. It outputs the serial numbers of the devices the ball entered in order to a file named `log.txt`.

Third, the sample grader prints the evaluation of your answer to the standard output.

- If your program is judged as **`Accepted`**, the sample grader prints S and P in the following format `Accepted: S P`.

- If your program is judged as **`Wrong Answer`**, it prints `Wrong Answer: MSG`. The meaning of `MSG` is as follows:

  - `answered not exactly once`: The procedure `answer` is called not exactly once.
  - `wrong array length`: The length of `C` is not $M + 1$, or the lengths of `X` and `Y` are different.
  - `over 400000 switches`: S is larger than $400\,000$.
  - `wrong serial number`: There is an element of `C`, `X`, or `Y` which is smaller than $-S$ or larger than M.
  - `over 20000000 inversions`: The ball doesn't return to the origin within $20\,000\,000$ state changes of the switches.
  - `state 'Y'`: There is a switch whose state is 'Y' when the ball first returns to the origin.
  - `wrong motion`: The triggers which cause motions are different from the sequence A.

Note that the sample grader might not create `out.txt` and/or `log.txt` when your program is judged as `Wrong Answer`.