A - Sanitize Hands

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There is a bottle of disinfectant that can disinfect exactly M hands.

N aliens come one by one to disinfect their hands.
The i-th alien ($1 \leq i \leq N$) has H_i hands and wants to disinfect all of their hands once.

Determine how many aliens can disinfect all of their hands.
Here, even if there is not enough disinfectant left for an alien to disinfect all of their hands when they start, they will use up the remaining disinfectant.

Constraints

- $1 \leq N, M \leq 100$
- $1 \leq H_i \leq 100$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & M & & \\
H_1 & H_2 & \ldots & H_N
\end{array}
$$

Output

Print the number of aliens who can disinfect all of their hands.

Sample Input 1

```
5 10
2 3 2 5 3
```

Sample Output 1

```
3
```

The aliens disinfect their hands in the following steps:

- The first alien disinfects their two hands. The remaining disinfectant can disinfect $10 - 2 = 8$ hands.
- The second alien disinfects their three hands. The remaining disinfectant can disinfect $8 - 3 = 5$ hands.
- The third alien disinfects their two hands. The remaining disinfectant can disinfect $5 - 2 = 3$ hands.
- The fourth alien has five hands, but there is only enough disinfectant for three hands, so they use up the disinfectant without disinfecting all of their hands.

Thus, the first three aliens can disinfect all of their hands, so print 3.

Sample Input 2

```
5 10
2 3 2 3 5
```

Sample Output 2

```
4
```

Sample Input 3

```
1 5
1
```

Sample Output 3

```
1
```

All aliens can disinfect their hands.

B - Uppercase and Lowercase

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 200 points

Problem Statement

You are given a string S consisting of lowercase and uppercase English letters. The length of S is odd.
If the number of uppercase letters in S is greater than the number of lowercase letters, convert all lowercase letters in S to uppercase.
Otherwise, convert all uppercase letters in S to lowercase.

Constraints

- S is a string consisting of lowercase and uppercase English letters.
- The length of S is an odd number between 1 and 99, inclusive.

Input

The input is given from Standard Input in the following format:

```
S
```

Output

Print the string S after converting the letters according to the problem statement.

Sample Input 1

```
AtCoder
```

Sample Output 1

```
atcoder
```

The string `AtCoder` contains five lowercase letters and two uppercase letters. Thus, convert all uppercase letters in `AtCoder` to lowercase, which results in `atcoder`.

Sample Input 2

```
SunTORY
```

Sample Output 2

```
SUNTORY
```

The string SunTORY contains two lowercase letters and five uppercase letters. Thus, convert all lowercase letters in SunTORY to uppercase, which results in SUNTORY.

Sample Input 3

```
a
```

Sample Output 3

```
a
```

C - Sierpinski carpet

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 250 points

Problem Statement

For a non-negative integer K, we define a level-K carpet as follows:

- A level-0 carpet is a 1×1 grid consisting of a single black cell.
- For $K > 0$, a level-K carpet is a $3^K \times 3^K$ grid. When this grid is divided into nine $3^{K-1} \times 3^{K-1}$ blocks:
  - The central block consists entirely of white cells.
  - The other eight blocks are level-$(K-1)$ carpets.

You are given a non-negative integer N.

Print a level-N carpet according to the specified format.

Constraints

- $0 \leq N \leq 6$
- N is an integer.

Input

The input is given from Standard Input in the following format:

```
N
```

Output

Print 3^N lines.

The i-th line ($1 \leq i \leq 3^N$) should contain a string S_i of length 3^N consisting of `.` and `#`.

The j-th character of S_i ($1 \leq j \leq 3^N$) should be `#` if the cell at the i-th row from the top and j-th column from the left of a level-N carpet is black, and `.` if it is white.

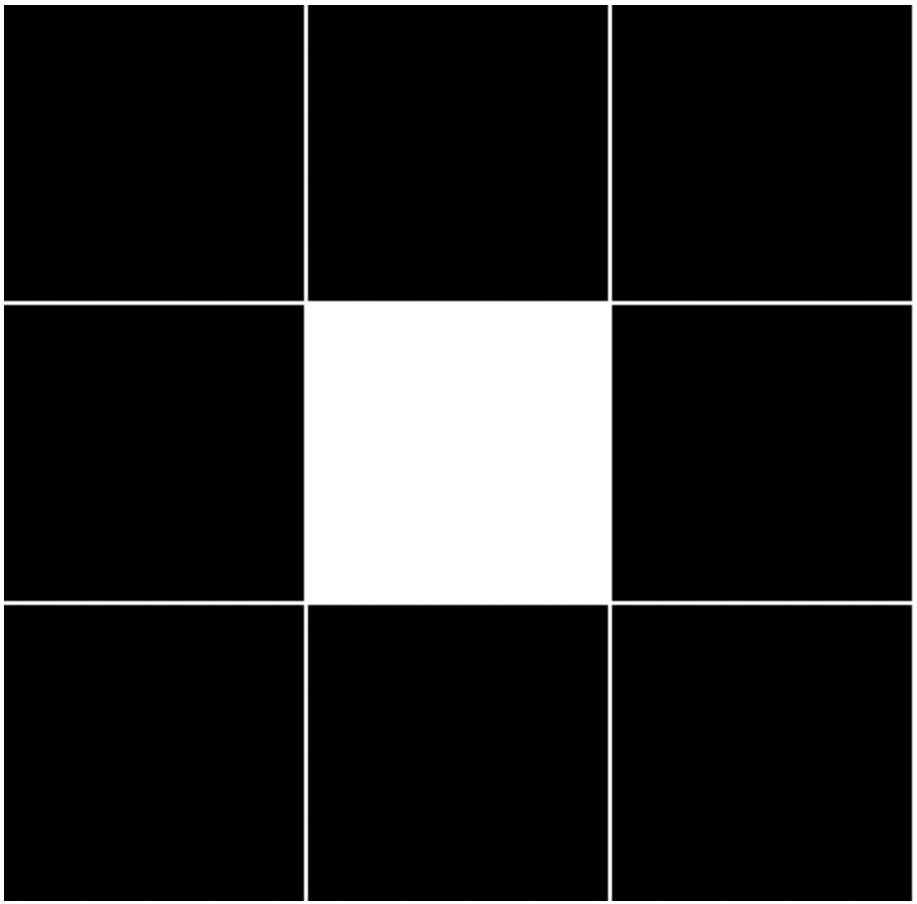
Sample Input 1

```
1
```

Sample Output 1

```
###
#.#
###
```

A level-1 carpet is a 3×3 grid as follows:



When output according to the specified format, it looks like the sample output.

Sample Input 2

```
2
```

Sample Output 2

```
#########
#.##.##.#
#########
###...###
#.#...#.#
###...###
#########
#.##.##.#
#########
```

A level-2 carpet is a 9×9 grid.

D - 88888888

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 350 points

Problem Statement

For a positive integer N, let V_N be the integer formed by concatenating N exactly N times.

More precisely, consider N as a string, concatenate N copies of it, and treat the result as an integer to get V_N.

For example, $V_3 = 333$ and $V_{10} = 10101010101010101010$.

Find the remainder when V_N is divided by 998244353.

Constraints

- $1 \leq N \leq 10^{18}$
- N is an integer.

Input

The input is given from Standard Input in the following format:

```
N
```

Output

Print the remainder when V_N is divided by 998244353.

Sample Input 1

```
5
```

Sample Output 1

```
55555
```

The remainder when $V_5 = 55555$ is divided by 998244353 is 55555.

Sample Input 2

```
9
```

Sample Output 2

```
1755646
```

The remainder when $V_9 = 999999999$ is divided by 998244353 is 1755646.

Sample Input 3

```
10000000000
```

Sample Output 3

```
468086693
```

Note that the input may not fit into a 32-bit integer type.

E - Reachability in Functional Graph

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 450 points

Problem Statement

There is a directed graph with N vertices numbered 1 to N and N edges.

The out-degree of every vertex is 1, and the edge from vertex i points to vertex a_i.

Count the number of pairs of vertices (u, v) such that vertex v is reachable from vertex u.

Here, vertex v is reachable from vertex u if there exists a sequence of vertices $w_0, w_1, \dots, w_K$ of length $K + 1$ that satisfies the following conditions. In particular, if $u = v$, it is always reachable.

- $w_0 = u$.
- $w_K = v$.
- For every $0 \leq i < K$, there is an edge from vertex w_i to vertex w_{i+1}.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq a_i \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

N
$a_1 \quad a_2 \quad \ldots \quad a_N$

Output

Print the number of pairs of vertices (u, v) such that vertex v is reachable from vertex u.

Sample Input 1

```
4
2 1 1 4
```

Sample Output 1

```
8
```

The vertices reachable from vertex 1 are vertices $1, 2$.
The vertices reachable from vertex 2 are vertices $1, 2$.
The vertices reachable from vertex 3 are vertices $1, 2, 3$.
The vertex reachable from vertex 4 is vertex 4.
Therefore, the number of pairs of vertices (u, v) such that vertex v is reachable from vertex u is 8.
Note that the edge from vertex 4 is a self-loop, that is, it points to vertex 4 itself.

Sample Input 2

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

Sample Output 2

```
14
```

Sample Input 3

```
10
6 10 4 1 5 9 8 6 5 1
```

Sample Output 3

```
41
```

F - Two Sequence Queries

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 550 points

Problem Statement

You are given sequences of length N, $A = (A_1, A_2, \ldots, A_N)$ and $B = (B_1, B_2, \ldots, B_N)$.

You are also given Q queries to process in order.

There are three types of queries:

- `1 l r x` : Add x to each of $A_l, A_{l+1}, \ldots, A_r$.
- `2 l r x` : Add x to each of $B_l, B_{l+1}, \ldots, B_r$.
- `3 l r` : Print the remainder of $\displaystyle\sum_{i=l}^{r}(A_i \times B_i)$ when divided by 998244353.

Constraints

- $1 \leq N, Q \leq 2 \times 10^5$
- $0 \leq A_i, B_i \leq 10^9$
- $1 \leq l \leq r \leq N$
- $1 \leq x \leq 10^9$
- All input values are integers.
- There is at least one query of the third type.

Input

The input is given from Standard Input in the following format. Here, query_i $(1 \leq i \leq Q)$ is the i-th query to be processed.

$$
\begin{aligned}
&N \quad Q \\
&A_1 \quad A_2 \quad \dots \quad A_N \\
&B_1 \quad B_2 \quad \dots \quad B_N \\
&\mathrm{query}_1 \\
&\mathrm{query}_2 \\
&\vdots \\
&\mathrm{query}_Q
\end{aligned}
$$

Each query is given in one of the following formats:

$$1 \quad l \quad r \quad x$$

$$2 \quad l \quad r \quad x$$

$$3 \quad l \quad r$$

Output

If there are K queries of the third type, print K lines.
The i-th line $(1 \leq i \leq K)$ should contain the output for the i-th query of the third type.

Sample Input 1

```
5 6
1 3 5 6 8
3 1 2 1 2
3 1 3
1 2 5 3
3 1 3
1 1 3 1
2 5 5 2
3 1 5
```

Sample Output 1

```
16
25
84
```

Initially, $A = (1, 3, 5, 6, 8)$ and $B = (3, 1, 2, 1, 2)$. The queries are processed in the following order:

- For the first query, print $(1 \times 3) + (3 \times 1) + (5 \times 2) = 16$ modulo 998244353, which is 16.
- For the second query, add 3 to A_2, A_3, A_4, A_5. Now $A = (1, 6, 8, 9, 11)$.
- For the third query, print $(1 \times 3) + (6 \times 1) + (8 \times 2) = 25$ modulo 998244353, which is 25.
- For the fourth query, add 1 to A_1, A_2, A_3. Now $A = (2, 7, 9, 9, 11)$.
- For the fifth query, add 2 to B_5. Now $B = (3, 1, 2, 1, 4)$.
- For the sixth query, print $(2 \times 3) + (7 \times 1) + (9 \times 2) + (9 \times 1) + (11 \times 4) = 84$ modulo 998244353, which is 84.

Thus, the first, second, and third lines should contain 16, 25, and 84, respectively.

Sample Input 2

```
2 3
1000000000 1000000000
1000000000 1000000000
3 1 1
1 2 2 1000000000
3 1 2
```

Sample Output 2

```
716070898
151723988
```

Make sure to print the sum modulo 998244353 for the third type of query.

G - Stair-like Grid

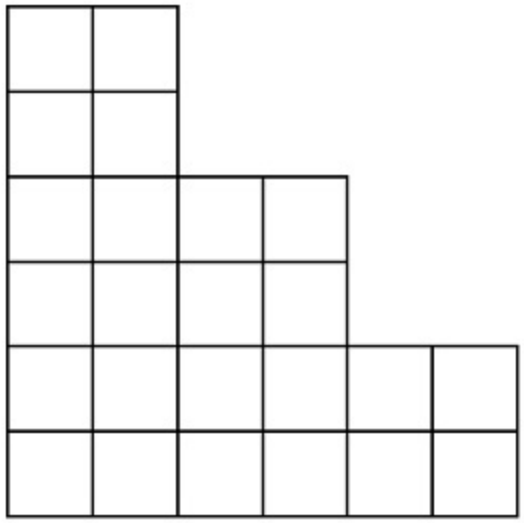
Time Limit: 6 sec / Memory Limit: 1024 MB

Score : 650 points

Problem Statement

There is a special grid with N rows. (N is even.) The i-th row from the top has $\left\lceil \frac{i}{2} \right\rceil \times 2$ cells from the left end.

For example, when $N = 6$, the grid looks like the following:



Let (i, j) denote the cell at the i-th row from the top and j-th column from the left.

Each cell is either an empty cell or a wall cell. There are M wall cells, and the i-th wall cell is (a_i, b_i). Here, $(1, 1)$ and (N, N) are empty.

Starting from $(1, 1)$, how many ways are there to reach (N, N) by repeatedly moving right or down to an adjacent empty cell? Find the count modulo 998244353.

Constraints

- $2 \leq N \leq 2.5 \times 10^5$
- N is even.
- $0 \leq M \leq 50$
- $1 \leq a_i \leq N$
- $1 \leq b_i \leq \left\lceil \frac{a_i}{2} \right\rceil \times 2$
- $(a_i, b_i) \neq (1, 1)$ and $(a_i, b_i) \neq (N, N)$.
- $(a_i, b_i) \neq (a_j, b_j)$ if $i \neq j$.
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{ll}
N & M \\
a_1 & b_1 \\
a_2 & b_2 \\
\vdots & \\
a_M & b_M
\end{array}
$$

Output

Print the number of ways to reach (N, N) from $(1, 1)$ by repeatedly moving right or down to an adjacent empty cell, modulo 998244353.

Sample Input 1

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

Sample Output 1

```
2
```

There are two paths that satisfy the conditions of the problem:

- $(1,1) \to (1,2) \to (2,2) \to (3,2) \to (3,3) \to (3,4) \to (4,4)$
- $(1,1) \to (1,2) \to (2,2) \to (3,2) \to (3,3) \to (4,3) \to (4,4)$

Sample Input 2

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

Sample Output 2

```
0
```

Sample Input 3

```
100 10
36 9
38 5
38 30
45 1
48 40
71 52
85 27
86 52
92 34
98 37
```

Sample Output 3

```
619611437
```

Sample Input 4

```
100000 10
552 24
4817 255
7800 954
23347 9307
28028 17652
39207 11859
48670 22013
74678 53158
75345 45891
88455 4693
```

Sample Output 4

```
175892766
```