A - ?UPC

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 100 points

Problem Statement

You are given a string S. Here, the first character of S is an uppercase English letter, and the second and subsequent characters are lowercase English letters.

Print the string formed by concatenating the first character of S and `UPC` in this order.

Constraints

- S is a string of length between 1 and 100, inclusive.
- The first character of S is an uppercase English letter.
- The second and subsequent characters of S are lowercase English letters.

Input

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

$$S$$

Output

Print the string formed by concatenating the first character of S and `UPC` in this order.

Sample Input 1

```
Kyoto
```

Sample Output 1

```
KUPC
```

The first character of `Kyoto` is `K`, so concatenate `K` and `UPC`, and print `KUPC`.

Sample Input 2

```
Tohoku
```

Sample Output 2

```
TUPC
```

B - Heavy Snake

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 200 points

Problem Statement

There are N snakes.

Initially, the thickness of the i-th snake is T_i, and its length is L_i.

The weight of a snake is defined as the product of its thickness and length.

For each integer k satisfying $1 \leq k \leq D$, find the weight of the heaviest snake when every snake's length has increased by k.

Constraints

- $1 \leq N, D \leq 100$
- $1 \leq T_i, L_i \leq 100$
- All input values are integers.

Input

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

$$
\begin{array}{l}
N \quad D \\
T_1 \quad L_1 \\
T_2 \quad L_2 \\
\vdots \\
T_N \quad L_N
\end{array}
$$

Output

Print D lines. The k-th line should contain the weight of the heaviest snake when every snake's length has increased by k.

Sample Input 1

```
4 3
3 3
5 1
2 4
1 10
```

Sample Output 1

```
12
15
20
```

When every snake’s length has increased by 1, the snakes' weights become $12, 10, 10, 11$, so print 12 on the first line.

When every snake’s length has increased by 2, the snakes' weights become $15, 15, 12, 12$, so print 15 on the second line.

When every snake’s length has increased by 3, the snakes' weights become $18, 20, 14, 13$, so print 20 on the third line.

Sample Input 2

```
1 4
100 100
```

Sample Output 2

```
10100
10200
10300
10400
```

C - Various Kagamimochi

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 300 points

Problem Statement

There are N mochi (rice cakes) arranged in ascending order of size. The size of the i-th mochi $(1 \leq i \leq N)$ is A_i.

Given two mochi A and B, with sizes a and b respectively, you can make one kagamimochi (a stacked rice cake) by placing mochi A on top of mochi B if and only if a is at most half of b.

You choose two mochi out of the N mochi, and place one on top of the other to form one kagamimochi.

Find how many different kinds of kagamimochi can be made.

Two kagamimochi are distinguished if at least one of the mochi is different, even if the sizes of the mochi are the same.

Constraints

- $2 \leq N \leq 5 \times 10^5$
- $1 \leq A_i \leq 10^9 \ (1 \leq i \leq N)$
- $A_i \leq A_{i+1} \ (1 \leq i < N)$
- All input values are integers.

Input

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

$$N$$
$$A_1 \quad A_2 \quad \cdots \quad A_N$$

Output

Print the number of different kinds of kagamimochi that can be made.

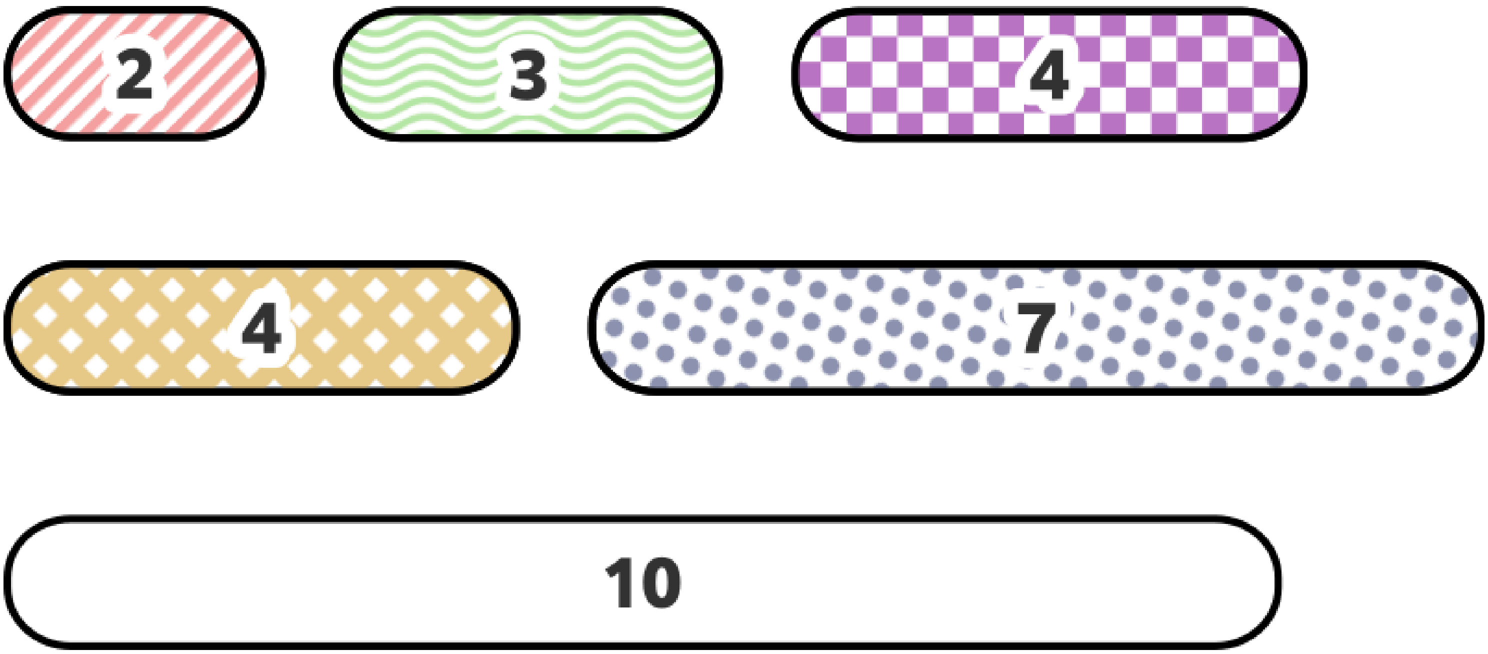
Sample Input 1

```
6
2 3 4 4 7 10
```

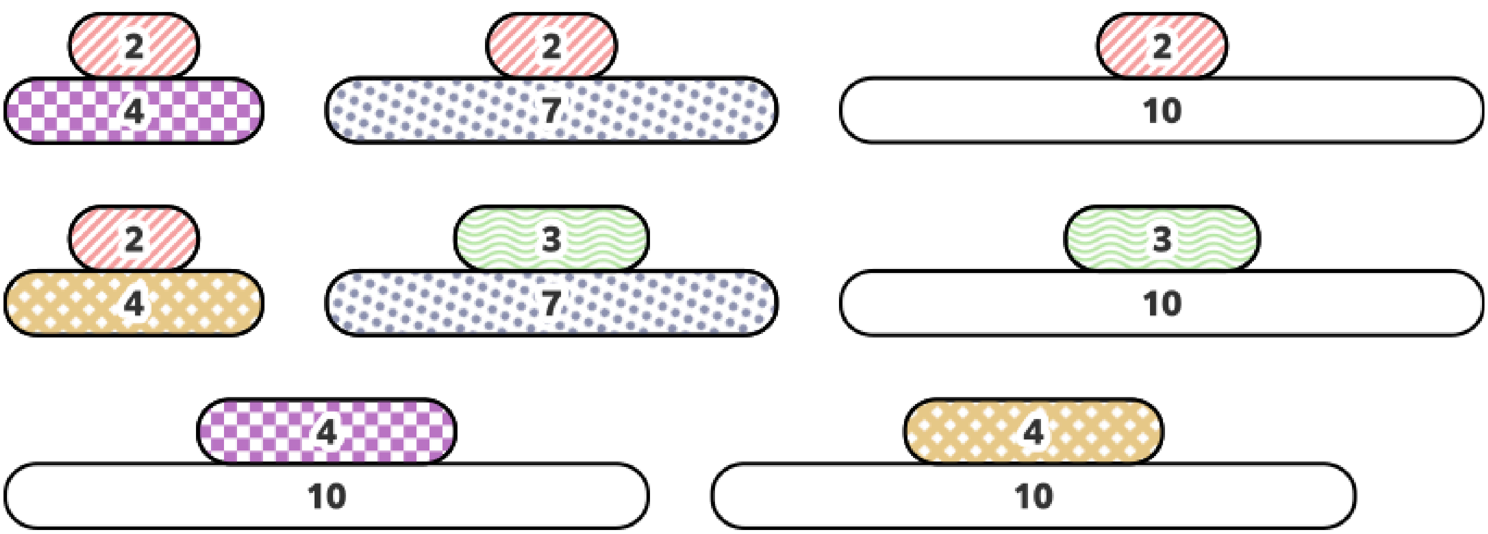
Sample Output 1

```
8
```

The sizes of the given mochi are as follows:



In this case, you can make the following eight kinds of kagamimochi:



Note that there are two kinds of kagamimochi where a mochi of size 4 is topped by a mochi of size 2, and two kinds where a mochi of size 10 is topped by a mochi of size 4.

Sample Input 2

```
3
387 388 389
```

Sample Output 2

```
0
```

It is possible that you cannot make any kagamimochi.

Sample Input 3

```
32
1 2 4 5 8 10 12 16 19 25 33 40 50 64 87 101 149 175 202 211 278 314 355 405 412 420 442 481 512 582 60
0 641
```

Sample Output 3

```
388
```

D - Coming of Age Celebration

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 400 points

Problem Statement

On a certain planet, there are N aliens, all of whom are minors.

The i-th alien currently has A_i stones, and will become an adult exactly i years later.

When someone becomes an adult on this planet, every **adult** who has at least one stone gives exactly one stone as a congratulatory gift to the alien who has just become an adult.

Find how many stones each alien will have after N years.

Assume that no new aliens will be born in the future.

Constraints

- $1 \leq N \leq 5 \times 10^5$
- $0 \leq A_i \leq 5 \times 10^5$
- 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

Let B_i be the number of stones owned by the i-th alien after N years. Print $B_1, B_2, \ldots, B_N$ in this order, separated by spaces.

Sample Input 1

```
4
5 0 9 3
```

Sample Output 1

```
2 0 10 5
```

Let C_i be the number of stones that the i-th alien has at a given time.

Initially, $(C_1, C_2, C_3, C_4) = (5, 0, 9, 3)$.

After 1 year, $(C_1, C_2, C_3, C_4) = (5, 0, 9, 3)$.

After 2 years, $(C_1, C_2, C_3, C_4) = (4, 1, 9, 3)$.

After 3 years, $(C_1, C_2, C_3, C_4) = (3, 0, 11, 3)$.

After 4 years, $(C_1, C_2, C_3, C_4) = (2, 0, 10, 5)$.

Sample Input 2

```
5
4 6 7 2 5
```

Sample Output 2

```
0 4 7 4 9
```

Sample Input 3

```
10
2 9 1 2 0 4 6 7 1 5
```

Sample Output 3

```
0 2 0 0 0 4 7 10 4 10
```

E - Simultaneous Kagamimochi

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 450 points

Problem Statement

There are N mochi (rice cakes), arranged in ascending order of size. The size of the i-th mochi $(1 \leq i \leq N)$ is A_i.

Given two mochi A and B, with sizes a and b respectively, you can make one kagamimochi (a stacked rice cake) by placing mochi A on top of mochi B if and only if a is at most half of b.

Find how many kagamimochi can be made simultaneously.

More precisely, find the maximum non-negative integer K for which the following is possible:

- From the N mochi, choose $2K$ of them to form K pairs. For each pair, place one mochi on top of the other, to make K kagamimochi.

Constraints

- $2 \leq N \leq 5 \times 10^5$
- $1 \leq A_i \leq 10^9 \ (1 \leq i \leq N)$
- $A_i \leq A_{i+1} \ (1 \leq i < N)$
- 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 maximum K such that K kagamimochi can be made simultaneously.

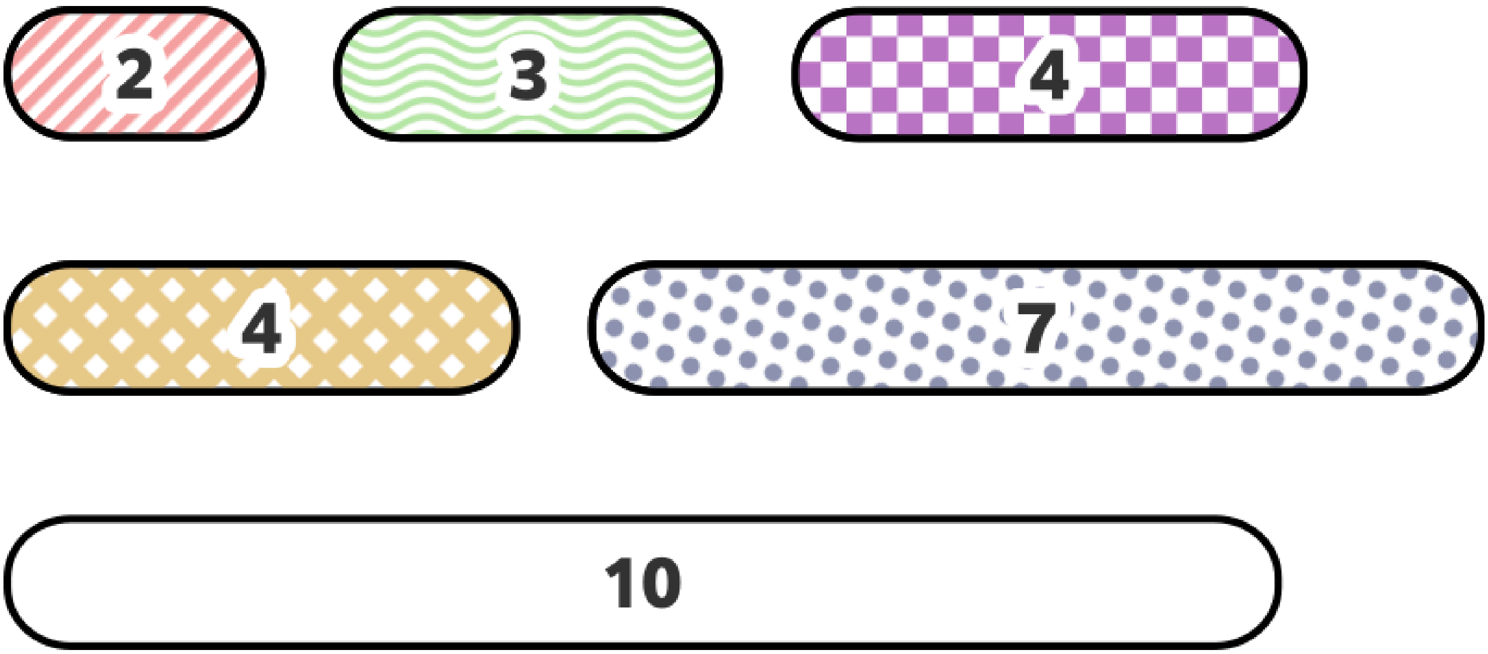
Sample Input 1

```
6
2 3 4 4 7 10
```

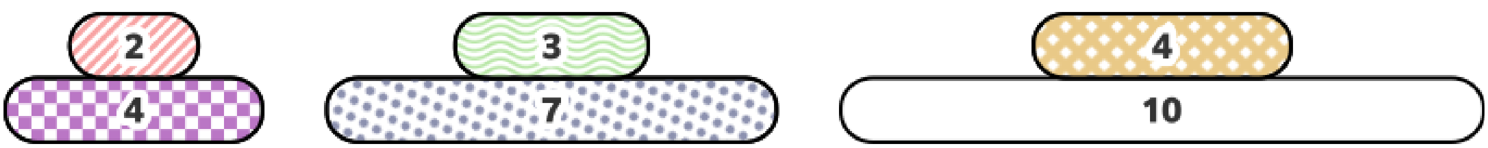
Sample Output 1

```
3
```

The sizes of the given mochi are as follows:



In this case, you can make the following three kagamimochi simultaneously:



It is not possible to make four or more kagamimochi from six mochi, so print 3.

Sample Input 2

```
3
387 388 389
```

Sample Output 2

```
0
```

It is possible that you cannot make any kagamimochi.

Sample Input 3

```
24
307 321 330 339 349 392 422 430 477 481 488 537 541 571 575 602 614 660 669 678 712 723 785 792
```

Sample Output 3

```
6
```

F - Dangerous Sugoroku

Time Limit: 4 sec / Memory Limit: 1024 MiB

Score : 550 points

Problem Statement

There are N squares arranged in a row, labeled $1, 2, \ldots, N$ from left to right.

You are given M pairs of integers $(L_1, R_1), \ldots, (L_M, R_M)$.

A square j is defined to be **bad** if and only if there exists some i such that $L_i \leq j \leq R_i$.

Determine whether you can move from square 1 to square N by repeatedly performing the following action:

- Let your current square be x. Choose an integer i that satisfies all of the following conditions, and move to square $x + i$.
  - $A \leq i \leq B$
  - $x + i \leq N$
  - Square $x + i$ is not bad.

Constraints

- $2 \leq N \leq 10^{12}$
- $0 \leq M \leq 2 \times 10^4$
- $1 \leq A \leq B \leq 20$
- $1 < L_i \leq R_i < N \ (1 \leq i \leq M)$
- $R_i < L_{i+1} \ (1 \leq i \leq M - 1)$
- All input values are integers.

Input

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

$$
\begin{aligned}
&N \quad M \quad A \quad B \\
&L_1 \quad R_1 \\
&L_2 \quad R_2 \\
&\vdots \\
&L_M \quad R_M
\end{aligned}
$$

Output

If it is possible to reach square N by repeating the action described in the problem statement, print `Yes`.

Otherwise, print `No`.

Sample Input 1

```
24 2 3 5
7 8
17 20
```

Sample Output 1

```
Yes
```

You can move to square N in this way: $1 \to 6 \to 9 \to 12 \to 16 \to 21 \to 24$.

Sample Input 2

```
30 1 5 8
4 24
```

Sample Output 2

```
No
```

Sample Input 3

```
100 4 10 11
16 18
39 42
50 55
93 99
```

Sample Output 3

```
Yes
```

G - Simultaneous Kagamimochi 2

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 575 points

Problem Statement

There are N mochi (rice cakes), arranged in ascending order of size. The size of the i-th mochi $(1 \leq i \leq N)$ is A_i.

Given two mochi A and B, with sizes a and b respectively, you can make one kagamimochi (a stacked rice cake) by placing mochi A on top of mochi B if and only if a is at most half of b.

You are given Q integer pairs. Let (L_i, R_i) be the i-th pair $(1 \leq i \leq Q)$, and solve the following problem for each i:

> Using only the $R_i - L_i + 1$ mochi from the L_i-th to the R_i-th, how many kagamimochi can you make simultaneously?
>
> More precisely, find the maximum non-negative integer K such that:
>
> - Out of the $R_i - L_i + 1$ mochi from the L_i-th to the R_i-th, choose $2K$ mochi and form K pairs. For each pair, place one mochi on top of the other, to make K kagamimochi.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq A_i \leq 10^9 \ (1 \leq i \leq N)$
- $A_i \leq A_{i+1} \ (1 \leq i < N)$
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq L_i < R_i \leq N \ (1 \leq i \leq Q)$
- All input values are integers.

Input

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

$$
\begin{array}{l}
N \\
A_1 \quad A_2 \quad \ldots \quad A_N \\
Q \\
L_1 \quad R_1 \\
L_2 \quad R_2 \\
\vdots \\
L_Q \quad R_Q
\end{array}
$$

Output

Print Q lines. The i-th line $(1 \leq i \leq Q)$ should contain the answer to the i-th query.

Sample Input 1

```
11
1 1 2 3 4 4 7 10 11 12 20
5
2 5
3 8
7 11
1 2
1 11
```

Sample Output 1

```
2
3
1
0
5
```

The answers to each query are as follows. One possible way to make the kagamimochi is given for each query.

- The mochi sizes are $(1, 2, 3, 4)$. You can make the two kagamimochi $(1, 3)$ and $(2, 4)$.
- The mochi sizes are $(2, 3, 4, 4, 7, 10)$. You can make the three kagamimochi $(2, 4)$, $(3, 7)$, and $(4, 10)$.
- The mochi sizes are $(7, 10, 11, 12, 20)$. You can make one kagamimochi $(10, 20)$.
- The mochi sizes are $(1, 1)$. You cannot make any kagamimochi.
- The mochi sizes are $(1, 1, 2, 3, 4, 4, 7, 10, 11, 12, 20)$. You can make five kagamimochi $(1, 2)$, $(1, 3)$, $(4, 10)$, $(4, 11)$, and $(7, 20)$.

Hence, print `2`, `3`, `1`, `0`, `5` in this order.

Sample Input 2

```
24
127 148 170 174 258 311 331 414 416 436 517 523 532 587 591 638 660 748 760 776 837 857 972 984
15
7 11
8 9
8 13
12 15
9 23
1 17
8 12
1 5
6 17
3 7
12 19
13 18
7 22
1 12
14 15
```

Sample Output 2

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