# GPC-UNI Resto del Mundo - Contest 03

March 4th, 2020

# Problem A. Good Dominoes

| | |
|---|---|
| Author: | racsosabe |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 4 seconds |
| Memory limit: | 256 megabytes |

Dominoes are $1 \times 2$ tiles which contain two integers in each side, say $a_i$, $b_i$.

The *digital sum* of a number $x$ in base $b \geq 2$ is the sum of its digits when it's written in base $b$.

The *digital root* of a number $x$ in base $b \geq 2$ is computed by repeatedly replacing it by its digital sum in base $b$ until there is only a one-digit number in base $b$. We denote it by $f_b(x)$.

For example, the digital root of 40 in base 3 is 2, since:

$$40 = \overline{1111}_{(3)} \to 4 = \overline{11}_{(3)} \to 2 = \overline{2}_{(3)}$$

We say that a pair $(k, b)$ is **good** if and only if:

$$\{f_b(i \cdot k), \forall i = 0, 1, \ldots, (b-1)\}$$

is a permutation of $(0, 1, \ldots, (b-1))$.

Artem has $n$ dominoes, he is allowed to rotate a domino $(a_i, b_i)$ to $(b_i, a_i)$.

Your task is to compute the maximum number of good pairs that can be obtained by possibly rotating some dominoes and get any optimal distribution of the dominoes.

## Input

The first line of input contains an integer $n$ ($1 \leq n \leq 3 \cdot 10^5$) — The number of dominoes Artem has.

The following $n$ lines of input contain two integers $a_i$ and $b_i$ ($1 \leq a_i, b_i \leq 10^{15}$) — The $i$-th domino has the values $a_i$ and $b_i$ on its sides.

## Output

In the first line, print the maximum number of good pairs that can be obtained.

Then, print $n$ lines with the optimal distribution of the dominoes:

In the $i$-th line, print the values $a'_i$ and $b'_i$, which should the result of rotating (or possibly not) the values $a_i$ and $b_i$.

**Warning:** Remember that the judge will only check if $(a'_i, b'_i)$ is a good pair, so be careful with your output format.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (10 points):**

- $1 \leq a_i, b_i \leq 10$

**Subtask 2 (25 points):**

- $1 \leq a_i, b_i \leq 1000$

**Subtask 3 (65 points):**

- $1 \leq a_i, b_i \leq 10^{15}$

For all the subtasks, it will hold:

- $1 \leq n \leq 3 \cdot 10^5$

## Examples

| standard input | standard output |
|---|---|
| 2<br>10 7<br>7 10 | 2<br>7 10<br>7 10 |
| 1<br>4 8 | 1<br>8 4 |

## Note

For the first sample case, notice that $(10, 7)$ is not a good pair, since:

$$f_7(0) = 0$$
$$f_7(10) = 4$$
$$f_7(20) = 2$$
$$f_7(30) = 6$$
$$f_7(40) = 4$$
$$f_7(50) = 2$$
$$f_7(60) = 6$$

Which is not a permutation of $\{0, 1, 2, 3, 4, 5, 6\}$.

However, $(7, 10)$ is a good pair, since:

$$f_{10}(0) = 0$$
$$f_{10}(7) = 7$$
$$f_{10}(14) = 5$$
$$f_{10}(21) = 3$$
$$f_{10}(28) = 1$$
$$f_{10}(35) = 8$$
$$f_{10}(42) = 6$$
$$f_{10}(49) = 4$$
$$f_{10}(56) = 2$$
$$f_{10}(63) = 9$$

Which is a permutation of $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

# Problem B. Careful Testing

| | |
|---|---|
| Author: | racsosabe |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 5 seconds |
| Memory limit: | 512 megabytes |

racsosabe is very busy setting up the problems for the "GPC-UNI Resto del Mundo Contest - 03". This time he is testing the solutions for one of the problems.

The problem is about getting range sums in a static array (the values won't change during the queries). The one who proposed this problem was TISparta and his model solution uses, since he is a data structures lover, a data structure called *Fenwick Tree*.

Fenwick Tree answers a range query using as a subrouting the function *getSum(pos)*, which gets the sum of all values in the range of indices $[0, pos]$. Thus, the function *query(L,R)* returns *getSum(R) - getSum(L-1)*.

The pseudocode of the function *getSum* is the following:

```
long long getSum(int pos){
    pos += 1;
    long long res = 0LL;
    while(pos > 0){
        res += ft[pos];
        pos &= pos-1;
    }
    return res;
}
```

However, TISparta forgot how this function was and the used this one:

```
long long getSum(int pos){
    pos += 1;
    long long res = 0LL;
    while(pos > 0){
        res += ft[pos];
        pos -= 1<<(31 - __builtin_clz(pos));
    }
    return res;
}
```

Given the current values of the array $ft$ of size $(n+1)$ and all the queries that the model solution must answer, compute the number of incorrect answers among all the tests.

**You can assume that all the $ft_i$ values have been computed correctly.**

## Input

The first line of input contains an integer $n$ $(1 \leq n \leq 10^5)$ — The number of elements in the original array $a$, the size of the Fenwick Tree array $ft$ is $(n+1)$.

The second line contains $(n+1)$ integers $ft_i$ $(0 \leq |ft_i| \leq 10^9)$ — The $i$-th integer is the $i$-th value of the Fenwick Tree array.

The third line contains an integer $q$ $(1 \leq q \leq 10^5)$ — The number of queries that need to be analyzed.

The following $q$ lines contain two integers $l_i$ and $r_i$ $(0 \leq l_i \leq r_i < n)$ — The $i$-th line contains the limits of the $i$-th query, for which one must answer the sum of all the values $a_l, \ldots, a_r$ of the original array.

## Output

Print a single line — The number of incorrect queries using TISparta's code.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (40 points):**

- $1 \leq n, q \leq 10^4$

**Subtask 2 (60 points):**

- $1 \leq n, q \leq 10^5$

For all the subtasks it holds:

- $0 \leq |ft_i| \leq 10^9$

## Example

| standard input | standard output |
|---|---|
| 5<br>0 1 -9 2 4 7<br>3<br>0 3<br>2 4<br>4 4 | 2 |

# Problem C. Palindrome Third

| | |
|---|---|
| Author: | racsosabe |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Lord El-Melloi II is a magician detective. He was trying to solve a mistery about an encrypted message in the house of his pupil Gray. They found a string that only consisted of lowercase english letters.

They had to make a ritual such that the original string would be transformed in another one that only had letters $a$, $b$ and $c$. In addition, any two consecutive letters were different.

For the final step it was required that they determined a palindrome subsequence from the former string, but with the following condition:

- Let's denote the string as $s$ and the palindrome subsequence taken as $t$, then:

$$|t| \geq \left\lceil \frac{|s|}{3} \right\rceil$$

Help Lord El-Melloi II and Gray to find any valid subsequence to solve the mistery.

**Definition:** A subsequence $t$ of a string $s$ is a string that can be obtained by eliminating some positions of $s$ and concatenating the remaining characters from left to right. *abc* is a subsequence of *adbca*, but *abcd* is not.

## Input

The first line of input contains a string $s$ ($3 \leq |s| \leq 10^5$) — The final step string that consists only of letters $a$, $b$ and $c$.

## Output

Print a single line — A valid palindrome subsequence. It is guaranteed that there exists an answer.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (15 points):**

- $1 \leq |s| \leq 100$

**Subtask 2 (35 points):**

- $1 \leq |s| \leq 10^4$

**Subtask 3 (50 points):**

- $1 \leq |s| \leq 10^5$

## Example

| standard input | standard output |
|---|---|
| abc | c |

## Note

For the sample case, we need a palindrome subsequence of length at least 1, so the possible answers are:

- $a$
- $b$
- $c$

# Problem D. UNI Sports Week

| | |
|---|---|
| Author: | racsosabe |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Sports Week has come! There are $n$ Faculties (identified with ids from 1 to $n$) participating in a Volleyball Tournament. The Tournament is held using League Format, so each Faculty has a match against all others in the coliseum (so there is no home or away team) and the final standings are computed using the points gained and the sets won.

For simplicity (and since we don't have enough money to pay to the referees), our Volleyball matches are won by the best of 3 sets. Therefore, there are only 2 possible outcomes ($2-0$ and $2-1$ in sets). Finally, the winner is awarded with 2 points and the loser is awarded with only 1 point.

Our final standings will be according to the following priority:

1) Accumulated points

2) Ratio: $\frac{\text{Sets won}}{\text{Sets lost}}$

The **Championship** is given to all the teams **such that no other team has a strictly better result than them**.

There have been played $m$ matches so far, so you want to know how many different general results exist so that your team gets the Championship, just to decide if you will keep cheering it or not.

## Input

The first line of input contains 2 integers $n$, $w$ and $m$ $\left(2 \leq n \leq 16, 1 \leq w \leq n, 0 \leq \frac{n(n-1)}{2} - m \leq 10\right)$ — The number of Faculties, the id of your Faculty and the number of played matches so far.

The following $m$ lines contain 4 integers $a_i$, $b_i$, $pa_i$ and $pb_i$ $(1 \leq a_i, b_i \leq n, a_i \neq b_i, 0 \leq pa_i, pb_i \leq 2)$ — The ids from the Faculties that played the $i$-th match and the sets won by each of them, respectively.

## Output

Print a single line — "Just go home" if there isn't any general results that gives the championship to your faculty, else print the number of results that exist.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (100 points):**

- $2 \leq n \leq 16$
- $1 \leq w \leq n$
- $0 \leq \frac{n(n-1)}{2} - m \leq 10$

## Example

| standard input | standard output |
|---|---|
| 3 1 2<br>1 2 2 0<br>2 3 0 2 | 2 |

## Note

For the sample case, just notice that the current standings are:

| Faculty | Points | Sets Won | Sets Lost |
|---|---|---|---|
| 1 | 2 | 2 | 0 |
| 3 | 2 | 2 | 0 |
| 2 | 2 | 0 | 4 |

And there is only 1 match left, between 1 and 3, so if we want 1 to get the Championship, then there are only 2 results: winning $2-0$ or winning $2-1$.

# Problem E. XOR Emergence

| | |
|---|---|
| Author: | racsosabe |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 3 seconds |
| Memory limit: | 512 megabytes |

Knuckles has learned Bitwise XOR (^ in C++) in today's class, since he's really good at math he can get the answer for any set of numbers in no time. His teacher wanted to give him a challenge, so he gave him a set of distinct numbers and some queries. Then asked him to answer the queries with the sum of the Bitwise XOR of the elements for each subset of the query.

As you know de wae, he asked you to help him with this problem.

**Note:** Bitwise XOR of empty set is 0.

## Input

The first line of input will contain 2 integers $n$ and $q$ ($1 \leq n \leq 18, 1 \leq q \leq 10^5$) — The number of values and the number of queries.

The second line of input will contain $n$ integers $a_i$ ($0 \leq a_i \leq 10^9$) — The $i$-th integer is the $i$-th value.

The following $q$ line will contain a string $s$ ($|s| = n, s_i \in \{0, 1\}$) — The $i$-th character is 1 if the $i$-th number has to be considered in the query, 0 otherwise.

## Output

Print $q$ lines — The $i$-th line must contain the answer for the $i$-th query.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (30 points):**

- $1 \leq n \leq 9$

**Subtask 2 (70 points):**

- $1 \leq n \leq 18$

For all subtasks, it holds:

- $1 \leq q \leq 10^5$
- $0 \leq a_i \leq 10^9$

## Example

| standard input | standard output |
|---|---|
| 3 3<br>1 2 3<br>111<br>101<br>100 | 12<br>6<br>1 |

## Note

For the first query, we have $\{1, 2, 3\}$, which has the subsets $\{1, 2, 3\}$, $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$, $\{1\}$, $\{2\}$, $\{3\}$ and $\phi$, they sum up:

$$0 + 3 + 2 + 1 + 1 + 2 + 3 + 0 = 12$$

For the second query, we have $\{1, 3\}$, which has the subsets $\{1, 3\}$, $\{1\}$, $\{3\}$ and $\phi$, they sum up:

$$2 + 1 + 3 + 0 = 6$$

For the third query, we have $\{1\}$, which has the subsets $\{1\}$ and $\phi$, they sum up:

$$1 + 0 = 1$$

# Problem F. Inki

| | |
|---|---|
| Author: | Valgt |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 4 seconds |
| Memory limit: | 256 megabytes |

Inki is a hunter that is getting prepared for the winter, that's why he needs to store the maximum possible quantity of meat. In order to reach the hunting zone, he must cross a river and the only way to do so is by an old bridge composed by $n$ planks, the $i$-th of them has capacity $c_i$ (which is the maximum extra weight Inki can take with him without breaking it).

In addition, Inki has $m$ coins that he can use to repair some planks. He can repair the $i$-th plank and add 1 to its capacity for $k_i$ coins.

Help Inki to use his coins wisely, in order to get the maximum possible quantity of meat.

## Input

The first line of input contains two integers $n$ and $m$ ($1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 10^{12}$) — The number of planks in the bridge and the number of coins Inki has.

The second line contains $n$ integers $c_i$ ($1 \leq c_i \leq 10^5$) — The capacities of the $n$ planks.

The third line contains $n$ integers $k_i$ ($1 \leq k_i \leq 10^5$) — The cost of the $n$ planks, Inki can spend $k_i$ coins to repair the $i$-th plank and increment its capacity by 1.

## Output

Print a single line — The maximum quantity of meat that Inki can get.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (30 points):**

- $1 \leq n \leq 1000$
- $0 \leq m \leq 10^5$

**Subtask 2 (70 points):**

- $1 \leq n \leq 10^5$
- $0 \leq m \leq 10^{12}$

## Examples

| standard input | standard output |
|---|---|
| 5 12<br>1 5 10 6 4<br>3 2 2 1 1 | 4 |
| 3 12<br>1 1 1<br>1 2 3 | 3 |

## Note

For the first sample case, Inki can repair the first plank 3 times, turning its capacity into 4.

For the second sample case, Inki can repair each plank 2 times, turning their capacity into 3.

# Problem G. Sum of dices

| | |
|---|---|
| Author: | TISparta |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 3 seconds |
| Memory limit: | 256 megabytes |

TISparta is playing a game with dices. A dice is a cube, its faces contain distinct integers from 1 to 6 represented by black dots. In this game he will choose $n$ dices and a positive number $target$, then he will throw the $n$ dices and if the sum of the obtained numbers is $target$, he wins.

Moreover, TISparta considers that two plays are the same if the sorted numbers of the throws are the same. For example, if in one game with $n = 3, sum = 6$ he gets these values $(dice_1 = 3, dice_2 = 2, dice_3 = 1)$, he considers that it is the same as if he would have obtained these values $(dice_1 = 1, dice_2 = 2, dice_3 = 3)$ with the dices (since both give the sorted sequence $\{1, 2, 3\}$).

Now, he is wondering in how many ways he can win. Can you help him?

## Input

The only line of input contains two integers $n, m$ $(1 \leq n \leq 50, 1 \leq m \leq 6 \cdot n)$ — The number of dices and the target sum.

## Output

Print a single line — The number of ways TISparta can win.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (30 points):**

- $1 \leq n \leq 10$

**Subtask 2 (70 points):**

- $1 \leq n \leq 50$

For all subtasks, it holds:

- $1 \leq m \leq 6 \cdot n$

## Examples

| standard input | standard output |
|---|---|
| 1 6 | 1 |
| 2 4 | 2 |

## Note

For the first sample case, the only possible result is to get a 1.

For the second sample case, the only possible results are $\{1, 3\}$ and $\{2, 2\}$.

# Problem H. Divisor sequence

| | |
|---|---|
| Author: | TISparta |
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 3 seconds |
| Memory limit: | 1024 megabytes |

Leonidas is giving you this sequence:

$$f_0 = 0$$
$$f_n = f_{n-1} + \sigma_0(n)$$

Where $\sigma_0(n)$ is the number of divisors of $n$.

Now, he wants to know how much you have learned during this summer training. Then, he is giving you $Q$ queries, in each query you receive two positive numbers $l, r$ and for each query you must print how many prime numbers there are in $f_l, f_{l+1}, \ldots, f_r$.

## Input

The first line of input contains an integer $Q$ ($1 \leq Q \leq 10^5$) — The number of queries.

The following $Q$ lines of input contain two integers $l$ and $r$ ($1 \leq l \leq r \leq 10^6$) — The $i$-th query.

## Output

Print $Q$ integers — the $i$-th integer denotes the answer to the $i$-th query.

## Scoring

The scoring will be divided in the following subtasks:

**Subtask 1 (10 points):**

- $1 \leq Q \leq 10^3$
- $1 \leq l \leq r \leq 10^3$

**Subtask 2 (20 points):**

- $1 \leq Q \leq 10^4$
- $1 \leq l \leq r \leq 10^5$

**Subtask 3 (30 points):**

- $1 \leq Q \leq 10^5$
- $1 \leq l \leq r \leq 10^5$

**Subtask 4 (40 points):**

- $1 \leq Q \leq 10^5$
- $1 \leq l \leq r \leq 10^6$

## Examples

| standard input | standard output |
|---|---|
| 3<br>1 1<br>3 9<br>49 68 | 0<br>2<br>4 |
| 1<br>1 1000 | 132 |

## Note

The first $f_i$ are:

$$f_1 = 1$$
$$f_2 = 3$$
$$f_3 = 5$$
$$f_4 = 8$$
$$f_5 = 10$$
$$f_6 = 14$$
$$f_7 = 16$$
$$f_8 = 20$$
$$f_9 = 23$$
$$f_{10} = 27$$