**1.Problem Statement:**

While playing an RPG game, you were assigned to complete one of the hardest quests in this game. There are **n**monsters you’ll need to defeat in this quest. Each monster **i**is described with two integer numbers – **poweri**and **bonusi**. To defeat this monster, you’ll need at least **poweri**experience points. If you try fighting this monster without having enough experience points, you lose immediately. You will also gain **bonusi**experience points if you defeat this monster. You can defeat monsters in any order.

The quest turned out to be very hard – you try to defeat the monsters but keep losing repeatedly. Your friend told you that this quest is impossible to complete. Knowing that, you’re interested, what is the maximum possible number of monsters you can defeat?

**Input:**

* The first line contains an integer, n, denoting the number of monsters. The next line contains an integer, e, denoting your initial experience.
* Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer, poweri, which represents power of the corresponding monster.
* Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer, bonusi, which represents bonus for defeating the corresponding monster.

| **Input** | **Output** | **Output Description** |
| --- | --- | --- |
| 2 123 78 130 10 0 | 2 | * Initial experience level is 123 points. * Defeat the first monster having power of 78 and bonus of 10. Experience level is now 123+10=133. * Defeat the second monster. |
| 3 100 101 100 304 100 1 524 | 2 | * Initial experience level is 100 points. * Defeat the second monster having power of 100 and bonus of 1. Experience level is now 100+1=101. * Defeat the first monster having power of 101 and bonus of 100. Experience level is now 101+100=201. * The third monster can’t be defeated. |

**2.Problem Statement:**

Your birthday is coming soon and one of your friends, Alex, is thinking about a gift for you. He knows that you really like integer arrays with interesting properties.

He selected two numbers, **N**and **K**and decided to write down on paper all integer arrays of length **K**(in form **a[1], a[2], …, a[K]**), where every number **a[i]**is in range from **1**to **N**, and, moreover, **a[i+1]**is divisible by **a[i]***(where 1 <****i****<=****K****)*, and give you this paper as a birthday present.

Alex is very patient, so he managed to do this. Now you’re wondering, how many different arrays are written down on this paper?

Since the answer can be really large, print it **modulo 10000**.

**Input:**

* The first line contains an integer, n, denoting the maximum possible value in the arrays.
* The next line contains an integer, k, denoting the length of the arrays.

| **Input** | **Output** | **Output Description** |
| --- | --- | --- |
| 2 1 | 2 | The required length is 1, so there are only two possible arrays: [1] and [2]. |
| 2 2 | 3 | All possible arrays are [1, 1], [1, 2], [2, 2]. [2, 1] is invalid because 1 is not divisible by 2. |
| 3 2 | 5 | All possible arrays are [1, 1], [1, 2], [1, 3], [2, 2], [3, 3]. |

**3.Problem Statement:**

You have an array ***A***of ***N***integers ***A1 A2 .. An***. Find the longest increasing subsequence ***Ai1 Ai2 .. Ak***(1 <= k <= N) that satisfies the following condition:  
For every adjacent pair of numbers of the chosen subsequence ***Ai[x]***and ***Ai[x+1]***(1 < x < k***),***the expression( ***Ai[x]***& ***Ai[x+1]***) \* 2 < ( ***Ai[x]***| ***Ai[x+1]***) is true

**Note**: ‘&’ is the bitwise AND operation, ‘ | ‘ is the bit-wise OR operation

**Input:**

1. The first line contains an integer, N, denoting the number of elements in A.
2. Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing Ai.

**Sample cases:**

| **Input** | **Output** | **Output Description** |
| --- | --- | --- |
| 5 15 6 5 12 1 | 2 | One possible subsequence is: 5 12 |
| 6 9 17 2 15 5 2 | 2 | One possible subsequence is: 2 15 |
| 7 17 16 12 2 8 17 17 | 3 | One possible subsequence is: 2 8 17 |

**4.Problem Statement :**

 You have been given a string S of length N. The given string is a binary string which consists of only 0’s and ‘1’s. Ugliness of a string is defined as the decimal   number that this binary string represents.

**Example:**

* “101” represents 5.
* “0000” represents 0.
* “01010” represents 10.

 There are two types of operations that can be performed on the given string.

* Swap any two characters by paying a cost of A coins.
* Flip any character by paying a cost of B coins
* flipping a character means converting a ‘1’to a ‘0’or converting a ‘0’ to a ‘1’.

  Initially, you have been given coins equal to the value defined in CASH. Your task is to minimize the ugliness of the string by performing the above mentioned operations on it. Since the answer can be very large, return the answer modulo 10^9+7.

**Note:**

* You can perform an operation only if you have enough number of coins to perform it.
* After every operation the number of coins get deducted by the cost for that operation.

**Input Format**

* The first line contains an integer, N, denoting the number of character in the string
* The next line contains a string, S, denoting the the binary string
* The next line contains an integer, CASH, denoting the total number of coins present initially
* Next will contains an integer, A, denoting the cost to swap two characters.
* Then the next line contains an integer, B, denoting the cost to flip a character.

**Constraints**

* 1 <= N <= 10^5
* 1< len(S)<= 10^5
* 1<=CASH <=10^5
* 1<=A<=10^5
* 1<=B<=10^5

**Sample Input 1 :**

4  
1111  
7  
1  
2

**Sample Output 1 :**

  1

**Explanation:**

   3 flips can be used to create “0001” which represents 1.

**Sample Input 2:**

  6  
  111011  
  7  
  1  
  3

**Sample Output 2:**

  7

**Explanation:**

  First swap 0 with the most significant 1, then use flip twice first on index one and then on index two “111011”=>”0111111″=>”001111″=>”000111″ the value represented is 7.

**Sample Input 3:**

  6  
  111011  
  7  
  3  
  2

**Sample Output 3:**

  3

**Explanation:**

 Flip the 3 most significant characters to get “000011” : the value represented by this string is 3.N

**5.Problem Statement :**

Khaled has an array A of N elements. It is guaranteed that N is even. He wants to choose at most N/2 elements from array A. It is not necessary to choose consecutive elements.  Khaled is interested in XOR of all the elements he chooses. Here, XOR denotes the bitwise XOR operation.

**For example:**

* If A=[2,4,6,8], then khaled can choose the subset [2,4,8] to achieve XOR=(2 XOR 4 XOR 8)=14.

Khaled wants to maximize the XOR of all the elements he chooses. Your task is to help khaled to find the max XOR of a subset that he can achieve by choosing at most N/2 elements?

**Input format:**

* The first line contains an integer, N, denoting the number of elements in A.
* Each line i of the N subsequent lines(where 0<=i<=N) contains an integer describing Ai.

**Constraints**

* 1<=N<=120
* 1<=A[i]<=10^6

**Sample Input 1**

   2  
   1  
   2  
**Sample Output 1**   
   2

**Explanation:**

N=2,  A=[1,2] khaled can choose the subset[2]. The xor of the elements in the subset is 2. And the number of elements in the subset is 1 which is less than N/2.

**Sample Input 2**4  
1  
2  
4   
7

**Sample Output 2**

7

**Explanation:**

N=4,  A=[1,2,4,7] Khaled can choose the subset [7]. The xor of the elements in the subset is 7, and the number of elements in the subset is 1 which is less than N/2.

**6.Problem Statement :**

Wael is well-known for how much he loves the bitwise XOR operation, while kaito is well known for how much he loves to sum numbers, so their friend Resli decided to make up a problem that would enjoy both of them. Resil wrote down an array A of length N, an integer K and he defined a new function called  Xor- sum as follows

* Xor-sum(x)=(x XOR A[1])+(x XOR A[2])+(x XOR A[3])+…………..+(x XOR A[N])

Can you find the integer x in the range [0,K] with the maximum Xor-sum (x) value?

Print only the value.

**Input format**

* The first line contains integer N denoting the number of elements in A.
* The next line contains an integer, k, denoting the maximum value of x.
* Each line i of the N subsequent lines(where 0<=i<=N) contains an integer describing Ai.

**Constraints**

* 1<=N<=10^5
* 0<=K<=10^9
* 0<=A[i]<=10^9

**Sample Input 1**

1  
0  
989898

**Sample Output 1**

989898

**Explanation:**

Xor\_sum(0)=(0^989898)=989898

**Sample Input 2**

3  
7  
1  
6  
3

**Sample Output 2**

14

**Explanation**

Xor\_sum(4)=(4^1)+(4^6)+(4^3)=14.

**Sample Input 3**

4  
9  
7  
4  
0  
3

**Sample Output 3**

46

**Explanation:**

Xor\_sum(8)=(8^7)+(8^4) +(8^0)+(8^3)=46.

**7.Problem Statement :**

One of the first lessons IT students learn is the representation of natural numbers in the binary number system (base 2) This system uses only two digits, 0 and 1. In everyday life we use for convenience the decimal system (base 10) which uses ten digits, from 0 to 9. In general, we could use any numbering system.

Computer scientists often use systems based on 8 or 16. The numbering system based on K uses K digits with a value from 0 to K-1. Suppose a natural number M is given, written in the decimal system To convert it to the corresponding writing in the system based on K, we successively divide M by K until we reach a quotient that is less than K

The representation of M in the system based on K is formed by the final quotient (as first digit) and is followed by the remainder of the previous divisions**For example :**

* If M=122 and K=8, 122 in base 10= 172 in base 8 This means that the number
* In decimal system = 172 in octal system.
* 172 in base 8 = 1\*8^2 + 7\*8 + 2 = 122

You made the following observation in applying the above rule of converting natural numbers to another numbering system

* In some cases in the new representation all the digits of the number are the same. For example 63 in base 10= 333 in base 4

Given a number M in its decimal representation, your task is find the minimum base B such that in the representation of M at base B all digits are the same.

**Input Format**

* The first line contains an integer, M, denoting the number given

**Constraints**

* 1 <= M = 10^12

**Sample Input 1 :**

     41

**Sample Output 1 :**

    40

**Explanation :**

Here 41 in base 40. will be 11 so it has all digits the same, and there is no smaller base satisfying the requirements

**Sample Input 2 :**

    34430

**Sample Output 2 :**

   312

**Explanation :**

Here 34430 in base 312 will have all digits the same and there is no smaller base satisfying the requirements