

# CS F211

## Data Structures and Algorithms

### Assignment - 2

Allowed languages: C

January 29, 2021

### General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on [Online GDB](#). If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

## A: Ghot's Riddle

The students taking the C programming course recently learnt how to find the  $n^{th}$  number in the fibonacci sequence, but found it to be too easy. So they asked the class ghot to come up with a more difficult function to find. The relation he came up with is defined as follows:

$$f(x) = \begin{cases} f(div_1) + f(div_2) & \text{if } x \text{ has two or more distinct divisors other than 1 and itself} \\ x & \text{if } x \text{ has 1 divisor or less apart from 1 and itself} \end{cases}$$

Where  $div_1$  and  $div_2$  are the two largest distinct divisors of  $n$ , other than 1 and the number itself. In this question, you will be provided with a number  $n$  and your task is to find  $f(n)$ . *Note: this is **not** the Fibonacci sequence defined above. Please read the question.*

### Input

The first line of input contains a single integer  $N$  ( $1 \leq N \leq 10^5$ ), for which you must find  $f(N)$ .

### Output

The output should contain a single integer that is the answer as described by the function above.

---

input

10

output

7

explanation

$f(10) = f(5) + f(2)$ , Since 5 and 2 do not have any proper divisors,  $f(5) = 5$  and  $f(2) = 2$  and hence  $f(10) = 5 + 2$  which is equal to 7.

---

input

8

output

6

---

## B: Tourism

One day you decide to visit the Grand Line for some sightseeing. The grand line consists of  $N$  different islands numbered from 1 to  $N$ . The waters between the Grand Line are incredibly dangerous and it's easy to get lost, so you buy a log pose to help you navigate. Depending on the island you are currently standing on, the log pose points to zero or more other islands and you can travel to any of these islands. You can start your journey from any island initially. You want to know the maximum number of **distinct** islands that can be reached from a starting island of your choice (the island you start from should also be counted) *Note that from your current island you can only travel to an island that the log pose points to. Keep in mind that you need to find the maximum number of islands that you can possibly reach from the starting island, not a path with the largest number of islands. See sample case for clarity.*

### Input

The first line of input contains two space-separated integers  $N$  and  $M$  ( $1 \leq N \leq 1000$ ,  $0 \leq M \leq 1000$ ) denoting the number of islands and island transitions you can make respectively. In the  $M$  lines that follow, the  $i^{th}$  line containing two space-separated integers,  $U_i$  and  $V_i$  representing that from island  $U_i$ , one of the islands that the log pose points to is  $V_i$ .

### Output

Print a single integer  $X$  denoting the maximum number of islands that you can reach if you start from some island.

---

input

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

output

```
5
```

explanation

if we draw the corresponding possible paths between islands, we can see that from the island 1 we can reach all other islands. From 2, we can't reach any other node. from 3 we can reach 3 and 2. from 4 we can reach only itself. from 5 we can only reach 5 and 4. Clearly we can see that if we start from node 1 our answer would be 5 which is maximum possible.

---

## C: Klutz

$N$  people are participating in the binary blitz competition this time, each numbered from 1 to  $N$  and all of them have a particular rating on codeforces. It is known that if two people compete, the one with the higher rating wins for sure. It is guaranteed that the ratings of all the  $N$  people are unique. The problem is that no one provided their actual codeforces handle so you don't know anyone's rating. All the  $N$  people play exactly once against every other person. So obviously there will be a total of  $\frac{N(N-1)}{2}$  games in total played by the  $N$  people. The competition is over and you have kept a record of all the matches in the following format: You have  $\frac{N(N-1)}{2}$  rows, and each row containing two space separated integers. The first number of these is the winner and the second one is the loser. It turns out you have lost exactly one row from this list and you have to try and fill in the missing record so that it doesn't contradict any of the other rows. If it is not possible to determine the winner and loser, print them in any order.

### Input

The first line contains a single integer  $N$  which is the number of people participating ( $3 \leq N \leq 200$ ). The following  $\frac{N(N-1)}{2} - 1$  lines contain two space-separated integers each  $U_i$  and  $V_i$ , where  $U_i$  is the winner of that round and  $V_i$  is the loser. *As an extra challenge, try to solve this with the constraint  $3 \leq N \leq 2000$ .*

### Output

Output a single line containing two space-separated integers that is the missing entry. If you cannot determine the winner and loser with the give information then print them in any order

---

input

4  
4 2  
4 1  
2 1  
3 1  
2 3

output

4 3

explanation

notice that if we put 3 4 then it would contradict line 1 and line 5

---

## D: Geass Code

You are the leader of the resistance against Britannia and you have  $N$  Knightmares that you can use during the revolution. The  $i^{th}$  Knightmare initially has a power  $P_i$ . You decide that the current power of your Knightmares isn't enough so you ask your lead scientist Rakshata to help you out. She comes up with  $M$  power multipliers placed one after the other in a row. each multiplier has a different power where  $i^{th}$  multiplier has a power  $Mult_i$ . She uses this in the following way: Start by placing the first index of the multiplier over the first element in  $P$ . Then for all  $1 \leq i \leq M$ ,  $P_i = P_i \cdot Mult_i$ . Then move the starting position of the multiplier to the second position of  $P$ . Then for all  $1 \leq i \leq M$ ,  $P_{i+1} = P_{i+1} \cdot Mult_i$ . Then move the starting position of the multiplier over to the third element of  $P$  and for all  $i$   $1 \leq i \leq M$ ,  $P_{i+2} = P_{i+2} \cdot Mult_i$ . Keep moving the starting position of the multiplier and multiplying its elements in a similar manner. Do this process a total of  $n - m + 1$  times. You must output the final powers of all the knightmares after all the operations are completed. Since the numbers can be very large, output them modulo  $(10^9 + 7)$ . *Note: Please ensure to use long int/long long int to avoid integer overflow since upon multiplication, the numbers can exceed the size of int.*

### Input

The first line consists of two space-separated integers  $N$  and  $M$ , the number of Knightmares and the number of multipliers. ( $1 \leq M \leq N \leq 10^5$ ). The second line contains  $N$  space-separated integers which are the initial powers of the knightmares  $P_1, P_2, \dots, P_n$ . The third line contains  $M$  space-separated integers, representing the power of each of the  $M$  multipliers,  $A_1, A_2, \dots, A_m$  ( $1 \leq P_i, A_i \leq 10^9 \forall i$ ).

### Output

Print a single line of  $N$  space-separated integers, the final powers of the knightmares after all operations modulo  $(10^9 + 7)$ .

---

input

5 3  
5 3 1 7 2  
1 2 4

output

5 6 8 56 8

explanation

Initially the array  $A$  gets multiplied on to the first three elements of  $P$ , then the second three, and so on. It is easy to see that this is the answer.

---

## E: Who Will Win Today?

Shirogane wants to impress Kaguya and so he tells her that given a set of letters from the english alphabet, he can enumerate all possible words of length  $K$  using only those letters in 1 second or less. Obviously, this is an impossible task so he asks you, Ishigami, to bail him out so that he doesn't look stupid in front of Kaguya.

### Input

The first line consists of two space-separated integers  $N$  and  $K$ , the number of letters you can use, and the length of each word respectively (it is guaranteed that  $N^K \leq 10^5$  and sum of lengths of all possible words does not exceed  $10^5$ ). The second line contains a string of  $N$  **distinct** lowercase English alphabets (no capital letters or special characters) representing the possible letters that you can use. Note that you cannot use any character or letter not present in the string.

### Output

print  $N^K$  lines, each line containing a distinct string of length  $K$ . Two strings are considered distinct if they differ in at least one position.

---

input

2 3

ab

output

aab

bbb

abb

bab

baa

aba

bba

aaa

---

## F: Koro-sensei and the Powers of Two

While the students of class 3E thought they bought some respite after last week's problem on primes, Koro-sensei is back at it again, with another weird question. Given a number  $N$ , the students have to find the *number of ways* it can be represented as a sum of power of 2. Can you help them?

### Input

The only line of input contains a single integer  $N$ . ( $0 \leq N \leq 10^2$ )

### Output

Print one integer,  $X$  denoting the number of ways  $N$  can be expressed as a sum of powers. Please note that reorderings of the same sum do not count as multiple ways - eg.  $(2+2+1)$ ,  $(2+1+2)$ ,  $(1+2+2)$  are all treated as the same thing.

---

input

7

output

6

explanation

7 can be represented as:  $(4+2+1)$ ,  $(4+1+1+1)$ ,  $(2+2+2+1)$ ,  $(2+2+1+1+1)$ ,  $(2+1+1+1+1+1)$ , and  $(1+1+1+1+1+1+1)$ , since the powers of 2 are  $2^0, 2^1, 2^2 \dots$  which are 1, 2, 4...

---

input

4

output

4

explanation

$(4)$ ,  $(2+2)$ ,  $(2+1+1)$ ,  $(1+1+1+1)$

---

## G: The COVID Vaccine

It's finally here and all this can end! However, before normalcy can completely return, enough people need to be vaccinated for everything to be safe. The owner of a vaccine manufacturing facility wants to start selling and shipping vaccines as soon as possible. However, business being business, she wants to ensure that the company spends the *least* amount of money for shipping the products. Vaccines are first stored in small boxes, which are then stacked inside big boxes before being loaded onto cargo ships and planes. All big boxes have a fixed weight  $B$ , but the weights of the small boxes can vary considerably. Given a list of weights of  $N$  small boxes, what is the *minimum* number of big boxes you'd need to ship all the vaccines available? **At most two boxes  $(s_i, s_j)$  can be placed in one big box**, due to government regulations.  $s_i + s_j \leq B$ . *Additional thinking: How would the answer change if this government regulation was removed? To be clear this is **not** a part of the problem statement - just something to think about.*

### Input

The first line of input contains  $N$  ( $1 \leq N \leq 2 \times 10^3$ ). The second line of input contains a sequence of  $N$  space-separated integers ( $s_i \leq B \leq 10^4$ ), representing the weights of the small boxes that we have. The third line of input contains the number  $B$ , the weight of the large box.

### Output

Print a single integer  $X$  denoting the minimum number of large boxes you would need.

---

input

2  
2 1  
3

output

1

explanation

We can fit both these small boxes in  
one big box of size 3

input

10  
9 12 7 5 4 3 3 1 8 4  
13

output

5

explanation

One valid configuration: (12, 1),  
(9, 4), (8, 5), (7, 4), (3,3)

---



## H: Aggregating the Binary Tree

You are given a complete binary tree in which all the nodes are at the same depth relative to the root. The binary tree is represented as an array  $A$ , where  $A[0]$  is the root. For every  $i^{th}$  index, it's left child is stored in  $A[2i + 1]$  and its right child in  $A[2i + 2]$ . You need to aggregate all the elements in the binary tree following a special rule:  $S = \sum_i A_i \times L_i$ , where  $A_i$  is the value at that node and  $L_i$  is the level. *Note: the level of the root node is 1, and the level of it's children would be 2, it's grandchildren 3, and so on.*

### Input

The first line contains a single integer  $N$  ( $1 \leq N \leq 2^{18} - 1$ ) denoting the number of elements in the binary tree where  $N$  is of the form  $2^k - 1$  where  $k$  is a positive integer. The next line contains  $N$  positive space separated integers.

### Output

Print a single integer  $S$ , denoting the aggregate sum of the binary tree.

---

input

3  
21 9 7

output

53

explanation

The root of the binary tree is 21 (level 1), while it's left child is 9 (level 2) and its right child is 7 (level 2). Aggregated sum =  $(21 \times 1) + (9 \times 2) + (7 \times 2) = 53$

---

input

7  
1 2 4 6 7 5 3

output

76

explanation

$1 * 1 + (2 + 4) * 2 + (6 + 7 + 5 + 3) * 3 = 76$

---

# I: The Dinosaur Conundrum

The inauguration of the Jurassic Park is about to happen tomorrow, and John Hammond has a problem on his hands. While they could successfully clone  $N$  different species of dinosaurs, they soon realized that dinosaurs would start attacking each other as soon as they woke up. After a lot of observation, the handlers notice that not all dinosaurs are aggressive towards each other - only dinosaurs that have been cloned from the same raw genetic material fight each other. Each dinosaur has a tag, which indicates the source of their genetic material. If the tags of two dinosaurs start with the same first character, they fight, otherwise they don't. You have  $C$  different cages to store the dinosaurs in, each of which can store any number of dinosaurs. Given the tags of all  $N$  dinosaurs, separate the dinosaurs into cages such that a *minimum* number of pairs of dinosaurs fight. For eg, if a cage has three dinosaurs - ("XAHSG-2737", "XAGFS-89" and "XSGFS-999") - each dinosaur will fight with the other two to form 3 pairs of fights - ("XAHSG-2737", "XAGFS-89"), ("XAHSG-2737", "XSGFS-999"), ("XAGFS-89" and "XSGFS-999").

## Input

The first line of input contains two space-separated integers,  $N$  and  $C$ . The following  $N$  lines contain strings (of length  $\leq 20$ ) denoting the tags of all the dinosaurs. It is guaranteed that the characters in the tag will be from the set [A-Z,0-9].

## Output

Print one integer,  $P$ , the number of pairs of fights that will happen in the minimal case.

---

input	input
6 4	6 2
XGB189	FFT998
ABFG99	DFFT99
XFG983	BWFT36
BST535	BERT99
TIN846	BLIP87
XC67TP	BHIJ78
output	output
0	2

---

## J. Air Tickets

You have been invited to the Annual Computing Conference which is going to be held in Tokyo this year. Sadly, the organizers have declined to reimburse your flight tickets to the conference. So, you decide to take the flight path (a flight path may have multiple hops) to Tokyo which will cost you the least. Given the all the hops possible between the major cities of the world, find the cost of the minimum flight path. Assume that each hop costs you the same, i.e. \$100. *Note: You must use adjacency lists to solve this problem.*

### Input

The first line of input contains four space-separated integers  $N$  ( $1 \leq N \leq 5000$ ),  $M$  ( $0 \leq M \leq \frac{N(N-1)}{2}$ ),  $S$  ( $0 \leq S \leq N-1$ ) and  $T$  ( $0 \leq T \leq N-1$ ) denoting the number of cities, the number of hops, the city you live in and city you wish to reach (Tokyo) respectively. The following  $M$  lines contain two space-separated integers  $U_i$  and  $V_i$  ( $0 \leq U_i, V_i \leq N-1$ ) denoting an undirected hop between the cities  $U_i$  and  $V_i$ . It is guaranteed that atleast one flight path exists to Tokyo.

### Output

Print a single integer  $C$ , denoting the cost of the minimum cost flight path to Tokyo from your city.

---

input

```
10 13 0 8
0 1
0 2
2 1
2 3
3 1
3 4
4 5
5 6
5 7
4 7
7 9
9 5
8 7
```

output

```
500
```

explanation

As you can observe, the minimum cost flight involves the hops, 0-2-3-4-7-8 (this is one of the flights paths, but it has minimum cost).

---