DATA STRUCTURES AND ALGORITHMS

ASSIGNMENT-7
Language allowed: C

February 17, 2020

A. Masterchef

Masterchef Vikas Khanna has a very busy day at work, especially on festivals when he is bombared with various orders. All the orders he gets are marked with a *priority number*. When Vikas gets a series of orders, he starts preparing the order with the *highest priority* and meanwhile if he gets another order with higher priority, he puts the current order in hold and starts working on the new order, else the order will be put in the queue of orders. If an order has been put on hold, Vikas resumes it only after all orders that came after it with higher priority have been completed. As Vikas's assistant, it's your task to manage all the orders. Hence, given the arrival time and preparation time and priority numbers of all the orders, can you figure out the sequence in which the orders will be completed along their time completion? Assume that if an order is about to completed at time T and simultaneosly (at time T) another order with priority greater that the rest of the orders comes up, Vikas first finishes the current current order and immediately takes the incoming order (there is no delay here, both events occur at time T). Also assume that if two orders have the same priority, Vikas takes up the order that has minimum remaining preparation time. Note that higher integer values indicate higher priority number.

Input

The first line contains a single integer N ($1 \leq N \leq 10^5$), denoting the number of orders Vikas gets that day. Each of the following N lines contains four space-separated integers O_i ($1 \leq O_i \leq 10^5$), X_i ($1 \leq X_i \leq 10^5$), A_i ($1 \leq A_i \leq 10^4$) and P_i ($1 \leq P_i \leq 10^4$) denoting the order number, priority number, arrival time and preparation time of the i^{th} order. It is given that the orders are numbered serially from 1 onwards. Note that the orders need not be in ascending order of arrival times.

Output

Print N lines, with each line containing two space-separated integers Y_j and T_j denoting the order number and completion time of j^{th} order. Note that these pairs must be printed in the sequence the orders the completed (see sample case for clarity).

```
input
6
1 2 5 3
2 1 5 3
3 6 3 6
4 1 2 2
5 5 7 5
6 3 3 4
```

```
output
3 9
5 14
6 18
1 21
4 22
```

B. Birthday Treat

It's Dr. Agasa's birthday and the detective boys have finally convinced him to give a party in the nearby restaurant. But due to the failure of his recent inventions, Dr. Agasa is short on money, and to avoid debts he wants the budget of the party to strictly under the money he currently has. As he doesn't want to disappoint the kids, he plans to buy as many dishes with the money he has. As he is bad at math, he turns to you, Edogawa Conan, to find which dishes he can buy for the kids. You must not use the inbuilt *qsort* function to solve this problem. You must solve this problem using *heaps* only. Note that each dish, say the i^{th} dish can be purchased only once.

Input

The first line of input contains two space-separated integers N ($1 \leq$ N $\leq 10^5$) denoting the number of the dishes in the restaurant and M ($0 \leq$ M $\leq 10^9$) denoting the amount of money Dr. Agasa currently posses. The following line contains N space-separated integers ($0 \leq P_i \leq 10^9$) denoting the prices of the dishes in the restaurant (the i^{th} integer denotes the price of the i^{th} dish).

Output

In the first line print a single integer K denoting the number of dishes Dr. Agasa can buy and in the following line print K space-separated integers denoting the indices of the dishes Dr. Agasa can buy. These indices must be printed in the order Dr. Agasa would buy the dishes. If more than one dish has the same price, Dr. Agasa prefers to buy the dish with the largest index first. If Dr. Agasa cannot buy any dish, just print "NO PARTY".

```
input
12 72
23 9 46 7 12 52 5 7 99 76 7 21

output
7
7 11 8 4 2 5 12
```

C. Heapify it!

In this task, you are given an array of integers and are expected to construct a *min-heap* out of those elements. After creating the heap, you need to sequentially pop all the elements and print them. Note that the pop operation implies the extract minimum element operation. Directly solving this problem by sorting and printing must not be done.

Input

The first line contains a single integer N ($1 \leq \text{N} \leq 2.10^5$), denoting the number of elements in the given array A. The next line contains N space-separated integers ($-10^9 \leq A_i \leq 10^9$) denoting the array itself.

Output

Print a single line containing N space-separated integers, denoting the integers in the sequence they are popped from the min-heap (the i^{th} integer in the output sequence should denote the element popped from the heap at the i^{th} instant).

```
input
13
8 -6 -56 90 12 -454 676 -129 -676 98 7 87 19

output
-676 -454 -129 -56 -6 7 8 12 19 87 90 98 676
```

D. Chocolates

Upon his return from the States, your father has brought you a lot of chocolates. On the first day, he arranges N bowls and adds a few chocolates to each bowl. Everyday (starting from day one), you decide to eat the chocolates from the bowls and select one bowl at random and eat X chocolates from it. The next day, your dad replenishes the bowl (from which you ate the chocolates) with $\lfloor X/3 \rfloor$ chocolates. You now start thinking about the *maximum* number of chocolates you can eat in a span of few days and decide to write a simple program that will calculate the same for you.

Input

The first line of input contains two space-separated integers N ($1 \leq N \leq 10^5$) and D ($1 \leq D \leq 10^5$) denoting the number of bowls and the number of days for which you want to make the calculation respectively. The following line contains N space-separated integers ($0 \leq A_i \leq 10^9$) denoting the number of chocolates put in each of bowls (the i^{th} integer denotes the number of chocolates put in the i^{th} bowl). Assume the bowls are numbered serially from 1 onwards.

Output

Print a single integer X, denoting the *maximum* number of chocolates you can eat by the end of D days. As the number can be large, print it to modulo $(10^9 + 7)$.

```
input
5 7
4 16 6 27 8

output
75
```

E. Find that Median!

In this task, you are given two arrays A and B each of size N. Both the array A and B will be sorted in ascending order. You now need to report the median of the array which is formed by merging A and B in ascending order. Write a program to do the same which has a complexity of *O(log N)*.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^6$) denoting the number of elements in each array. The next line contains N space-separated integers A_i ($-10^9 \leq A_i \leq 10^9$) denoting the first array A. The next line also contains N space-separated integers B_i ($-10^9 \leq B_i \leq 10^9$) denoting the second array B.

Output

Print a single integer/decimal M, denoting the median of the array formed by merging A and B in ascending order. If the merged array contains an even number of elements, print the average (floating point value with one place after the decimal point without any rounding) of both the integers. Make sure not to print extra trailing zeros.

```
input
5
1 11 13 17 19
4 8 12 15 18

output
12.5
```

```
input
4
1 5 9 10
1 7 10 11

output
8
```

F. Tree Traversals

Mafuyu Kirisu-sensei has yet another problem for Fumiya Furuhashi. This time, it is based on graphs. Kirisu gives Fumiya an *unweighted and undirected tree* T and asks her to print the level order traversal of the given tree T. As Kirisu felt that problem was too simple, she decided to slightly change the way of input. Help Fumiya code this problem. Assume that the tree is rooted at 0 and the input always constitutes a valid tree.

Input

The first line of input contains a single integer N ($1 \leq$ N $\leq 5.10^3$) denoting the number of vertices in the given tree T. Each of the following N lines contains a sequence of space-separated integers (note that, the number of integers is not given as input) A_i ($0 \leq A_i \leq$ N-1) denoting the vertices connected to the $(i-1)^{th}$ vertex. Assume that these N lines are numbered serially starting from 1.

Output

Print M (M = height(T)) lines with the i^{th} line containing the the vertex number of the vertices in the i^{th} level. The vertices in each level must printed in ascending order. (see sample case for clarity).

```
input
6
1 2
0
0 3 4
2
2 5
4

output
0
1 2
3 4
5
```

G. Kudo and the Book of Numbers

One day after class, Kudo finds a thick book laying on the pavement. On opening it, he finds a list of Roman numerals written inside. He is quite intrigued by it and wants to decipher those numbers, i.e. convert them to decimals numerals. But upon seeing list, he realizes that deciphering them correctly by hand will take him days. As Kudo has already promised Inspector Megure to help him with his upcoming investigation, he turns to you, Heiji, for help. Aid Kudo decipher all the Roman numerals in the book. The Roman Numerals consist of the letters I, V, X, L, C, D and M representing 1, 5, 10, 50, 100, 500 and 1000 respectively. Note that you must decipher those Roman Numerals using *subtractive notation* only. You are expected to know how Roman Numerals are constructed for solving this problem.

Input

The first line contains a single integer Q ($1 \le Q \le 10^5$) denoting the number Roman Numerals in the book. Each of the following N lines consists of a single string S ($1 \le |S| \le 10^3$, $S_i \in$ [I, V, X, L, C, D, M]) denoting the a Roman Numeral.

Output

Print Q lines (with the i^{th} line denoting the answer to the i^{th} query), with each line containing of a single integer M_i denoting the decimal equivalent of the i^{th} Roman Numeral query. If an input string does not represnt any Roman Numeral, print "INVALID_NUMBER". If the decimal equivalent of some Roman Numeral exceeds the range of **int**, print the value of INT_MAX for that corresponding query.

```
input
4
DCCXXVI
MMMDCCXXIV
VIIII
CCCC

output
726
3724
INVALID_NUMBER
INVALID_NUMBER
```

Use the link given below for referring to what *subtractive notation* is:
https://www.calculatorsoup.com/calculators/conversions/roman-numeral-calculator.php

H. Swaps

In this task, you are given an array A. You need to report the *minimum* number of swaps required to sort the array in *ascending* order. Write a simple program to do the same.

Input

The first line of input contains a single integer N ($1 \leq \mathrm{N} \leq 10^5$) denoting the number of elements in the array. The following line contains N space-separated integers ($-10^9 \leq A_i \leq 10^9$) denoting the given array A.

Output

Print a single integer X, denoting the *minimum* number of swaps required to sort the given array A in *ascending* order.

```
input
5
2 4 5 1 3

output
3
```

```
input
7
-5 -4 -1 6 8 10 12

output
0
```