CS F211
Data Structures and Algorithms
Assignment - 5

Allowed languages: **C**

February 17, 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: OS Scheduler

One of the most important tasks of an operating system is deciding which tasks gets to use the CPUs. Let us say that there are T tasks in the waiting list that require the CPU's computational power at time $t = 0$. No additional tasks will be added to the wait list of tasks. Each task requires some CPU time, E_i, for completion and has a priority, P_i (ranging from $0 - 10$, with 0 having the highest priority and 10 the least priority). The scheduling algorithm must work by (1) selecting *important tasks* first (as decided by P_i) and (2) in case of equal preference select the *shortest job first* (least time required). What are the indices of the first k tasks the scheduling algorithm prioritises? *Note: Your solution* ***must*** *be in* $o(T + k \cdot log(T))$.

Input

The first line contains two integers T ($1 \leq T \leq 10^5$) and k ($1 \leq k \leq 10^2$). The next N lines contain two space separated integers E_i ($0 \leq E_i \leq 10^9$) and P_i of the tasks (tasks are zero-indexed).

Output

Print k space separated integers representing the indices of first k tasks that will be scheduled by the OS.

```
input
8 3
34 3
2167 5
10 0
23 0
325 3
45 6
646 7
353 7

output
2 3 0

explanation
Schedule all tasks with priority 0 and then schedule the 0th task (priority 3).
```

B: Altered Carbon

In the Bay City Metropolis Hospital, N patients are waiting to be treated outside the hospital today. Being the dystopian society it is, the hospital wishes to *maximise* it's earning potential today. Each patient i has a certain *net wealth* C_i that is stored in an encrypted form in alphanumeric string D_i in their cortical stacks (an implant placed in the neck). The net worth C_i is the sum of all of the *alphabet-separated numbers* in the string. For example, the string "as3442jhs2323kaj22kjyu3yhdjhshdkjrhsjk211df" contains five numbers in it, and the net worth for this string would be the sum of these five numbers, i.e. $sum(3442, 2323, 22, 3, 211) = 6001$. Given that the hospital can admit only k ($1 \leq k << N$) patients, and wants to admit individuals with the highest net-worths, which ones should it select? What is the sum of these net worths? Since this number might be very large print it modulo $10^9 + 7$.

Input

The first line contains two space separated integers: N ($0 \leq N \leq 10^5$) and k ($0 \leq k \leq 100$). The next N lines contain one alphanumeric string each (of length ≤ 150) with each line containing one D_i.

Output

Print one integer representing the moduloed sums of net worths ($(\sum_i C_i)\%(10^9+7)$) of the selected individuals admitted in the hospital. Each string might contain alphabets (upper/lower case) and digits. It is guaranteed that each number inside the string will have at most 10 digits.

---

input
```
3 2
a21b45c90
a42b1d3FG6
98nahsg8
```

output
```
262
```

explanation
```
Person 0:  21 + 45 + 90 = 156
Person 1:  42 + 1 + 3 + 6 = 52
Person 2:  98 + 8 = 106
Select persons 0 and 2
```

input
```
5 3
a1ghe2
BGH54jk78
g3267372673hh7
jhj666SHH7
apes123strong23together9
```

output
```
267373487
```

explanation
```
The net worths are (2, 132,
3267372680, 673, 155).  The answer
is (3267372680 + 673 + 155)%(10^9 + 7).
```

C: Largest Number

Given two numbers N_1, N_2 (as digit strings) find the *largest* number you can create by following these steps. To merge the numbers:

1. start with an empty string M.
2. At each step you can choose to:
   (a) append the first digit of N_1 to the back of M and delete the first digit in N_1 (or)
   (b) append the first digit of N_2 to the back of M and delete the first digit in N_2
3. Keep repeating this process till both N_1 and N_2 are empty

What is the largest possible number you can create by merging N_1 and N_2.

Input

There will be two lines of input, with each line containing one very large number ($\leq$ 200 digits).

Output

Output a single line, containing the maximal merged number.

---

input
7756453241
82715243

output
877564532724152431

---

input
965
453

output
965453

---

input
775
769

output
777695

D: Valentine's Day

As an an unscrupulous store owner on a college campus, you decide to sell chocolates on Valentine's Eve in a bid to earn more money. Your supplier has M different kinds of chocolates all of which have different qualities (Q_i), and you want to select *exactly two* varieties of chocolates to stock. Since you intend to sell chocolates at a grossly overpriced sales price, you decide that you want to *maximise* the *difference* in qualities of the two selected chocolates - to make it *seem* like the chocolate of higher quality is worth the price relative to the lesser, even when both of them aren't worth it. Given a list of qualities of each type of chocolates, select the types that *maximise* this difference of qualities of chocolates. What is this maximal difference? And how many ways can you achieve this?

Input

The first line has the integer M ($2 \leq N \leq 10^6$). The next line has M space separated integers containing the *quality* of each type of chocolate ($0 \leq Q_i \leq 10^9$).

Output

Print two space-separated integers X and Y. X is the the largest possible difference of qualities of the two selected chocolate types. Y is the number of *ways* this maximum can be achieved.

```
input
2
9 278

output
269 1
```

```
input
3
19 23 24

output
5 1
```

```
input
5
24 15 19 15 24

output
9 4

explanation
There are 4 sets of indices that get the maxmimal difference of 9:  (0, 1),
(1, 4), (0, 3), (4, 0)
```

E: A Foreshadowing of Array Operations

You are given an array a of size N in non-decreasing order. In one operation you can choose any two adjacent elements and insert the floor of their average between them. This operation would increase the length of the array by 1. Let the maximum difference between any pair adjacent numbers in the array be m. Using at most k operations, find the minimum possible value of m that can be achieved after these operations.

As an extra challenge (will not be asked in the lab), try solving this if you can insert any integer such that the array remains non-decreasing.

Input

The first line contains two integers N and k ($1 \leq N, k \leq 10^5$). The second line contains N integers representing the array a ($1 \leq a_i \leq 10^9$).

Output

Print one integer, the minimum possible maximum difference between any two adjacent integers after all operations.

```
input
5 2
10 13 15 16 17

output
2

explanation
you can add up to 2 numbers.  Adding 11 and 14 in the appropriate positions would give
a maximum difference as 2.  It is not possible to do better in this situation
```

F: The Maxim

Migi the parasite is in a human host and trying to take over. To completely take over the host it must first take over all its organs which are numbered from 1 to N. You are given the pairs of organs that are connected to each other. Initially Migi is at organ 1. Migi can only take over a particular organ if it is not already occupied and it is connected to an occupied organ. The human host is fighting back pretty hard so Migi decides to do the following: **at every step choose the smallest numbered organ that is currently unoccupied and is connected to an occupied organ. Occupy the chosen organ**. Migi doesn't have much time so he asks you to tell him the exact order of organs to occupy including the starting organ.

Input

The first line consists of two integers N, M ($1 \leq N, M \leq 10^5$). The next M lines contain two integers u_i and v_i, indicating that organ u_i and organ v_i are connected ($1 \leq u_i, v_i \leq N$). It is guaranteed that all the organs form a connected graph.

Output

Print a single line consisting of N integers representing the order of organs Migi occupies (including the starting organ).

---

input

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

output

```
1 4 3 2 5
```

explanation

Initially Migi is at organ 1. Out of the connected organs 4 and 5, we choose 4 as it is the smallest. Now, 4 is connected to 3 and 1 is connected to 5, out of which 3 is the smallest so he occupies that. Now the remaining possible nodes are 2 and 5 so Migi goes to 2 first and then 5.

G: Array Operations

You are given an array of size N where the i^{th} element is a_i. In one operation you will remove the smallest and the largest elements in the array and insert the absolute value of their difference into the end of the array. If multiple elements are smallest or largest pick any one. You are given q queries, each of which consist of a single number k_i. For every i from 1 to q, you have to find the sum of elements after performing k_i operations on the **initial array** a.

Input

The first line contains two integers N and q ($1 \leq N, q \leq 10^5$). The second line contains N integers, representing the array a. The next q lines contain one integer each, representing the query k_i ($1 \leq k_i \leq N$),

Output

Output q lines, each line containing one integer representing the answer for the i^{th} query.

---

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

output
```
13
9
```

explanation

after the first operation, the array becomes 3 2 4 4. After the second operation the array becomes 3 4 2.

H: More Array Operations

After solving the previous problem, you wanted to challenge yourself some more so you decided to try a slightly different problem. You are given an array of size N where the i^{th} element is a_i. You have a new array b which is initially empty. In one operation you remove the leftmost element in a and add it somewhere in b. You do this N times until the array a is completely empty. After each step you have to print the median of all the elements in b. Median is defined in the following way: Let M be the size of array b. If M is even, then median is $b[\frac{M}{2}]$. Otherwise, the median is $\left\lfloor \frac{b[\frac{M-1}{2}]+b[\frac{M+1}{2}]}{2} \right\rfloor$. *Hint: Could you use heaps here?*

Input

The first line contains a single integer N representing the size of the array a. The second line contains N integers where the i^{th} integer is a_i.

Output

output N space separated integers where the i^{th} integer is the median of b after the i^{th} operation

---

input
4
5 15 1 3

output
5 10 5 4

explanation
first operation $\rightarrow b = [5]$. Second operation $\rightarrow b = [5, 15]$ so median is 10. Third move $b = [5, 15, 1]$ and here median is 5. In the final operation $b = [51513]$ where median is 4

I: 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 *all* X chocolates (assuming that the bowl has 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 Y, 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

explanation
It can be observed that you eat 27 + 16 + 9 + 8 + 6 + 5 + 4 = 75 chocolates in 7 days.
```

J. Dijkstra

Dijkstra's Single Source Shortest Path Algorithm is perhaps the most used procedure to find shortest paths in a weighted graph. The algorithm is greedy in nature and is optimally done with a min-priority queue (a heap). The pseudo-code for it is as follows:

1. Mark source vertex with 0 cost initially and rest all vertices with a cost of ∞.
2. With the cost of the source vertex, relax the cost of all its neighbours, i.e. `cost(v) = min(cost(`v`), cost(`u`) + W(`u`, `v`))` $\forall$ vertices v that are neighbors of source vertex u.
3. Mark u as visited and from those relaxed, choose the minimum cost vertex and make it the source vertex.
4. Go to step 2 and repeat till all vertices are visited.

Now given a weighted undirected connected graph G(V, E) and a source vertex $s \in$ G(V, E), your task is to find the shortest path from s to all the remaining vertices in the graph. *Additional: Can you guess the time complexity of Dijkstra using heaps?*

Input

The first line contains three space-separated integers N ($3 \leq$ N ≤ 500), M (N-1 $\leq$ M $\leq \frac{N(N-1)}{2}$) and S ($0 \leq$ S $\leq$ N-1) denoting the number of vertices, edges and source vertex respectively. The following M lines contain three space separated integers U_i, V_i and W_i ($0 \leq U_i, V_i \leq$ N-1, $0 \leq W_i \leq 10^7$) denoting an edge of weight W_i between vertices U_i and V_i.

Output

Print a sequence of N space-separated integers where the i^{th} integer denotes the shortest path distance to the i^{th} vertex from the given source vertex.

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

```
output
9 7 8 0 7 8 8

explanation
It can be observed by running
the algorithm that we get the
distances as mentioned above for
the vertices 0 to 6
```
