

# Palindrome Pairs

BETA Can't read the text? [Switch theme](#)

## 7. Complementary Pairs

A pair of strings form a complementary pair if there is some permutation of their concatenation that is a palindrome. For example, the strings "abac" and "cab" form a complementary pair since their concatenation is "abaccab" which can be rearranged to form a palindrome, i.e., "bcaaacb".

Given an array of  $n$  strings, find the number of complementary pairs that can be formed.

**Note:** Pairs of strings formed by indices  $(i, j)$  and  $(j, i)$  are considered the same.

**Example**  
Consider `stringData = ["abc", "abcd", "bc", "adc"]`.

The following complementary pairs can be formed:

- ("abc", "abcd"), concatenated string = "abcabcd" - arranged as a palindrome  $\rightarrow$  "abcdcba".
- ("abc", "bc"), concatenated string = "abcabc"  $\rightarrow$  "bcacb".
- ("abcd", "adc"), concatenated string = "abcdadc"  $\rightarrow$  "acdbdca".

Return 3, the number of complementary pairs.

**Function Description**  
Complete the function `countComplementaryPairs` in the editor below.

`countComplementaryPairs` has the following parameter:  
`string stringData[n]`: the strings to pair

**Returns**  
`long int`: the number of complementary pairs that can be formed

Language: `JavaScript`

```
20 * The function accepts STRING_A
21 */
22
23 public static long countComplementaryPairs(STRING_A[] list) {
24     // Write your code here
25     long ans = 0;
26     HashMap<Long, Long> map = new HashMap();
27     for(int i = 0; i < list.size(); i++) {
28         String cur = list.get(i);
29         long tmp = 0;
30         for(int j = 0; j < cur.length(); j++) {
31             tmp ^= (1 << (int)(cur.charAt(j) - 'a'));
32         }
33     }
34     ans += map.getOrDefault(0, 0);
```

**Test Results**    **Custom Input**

Compiled successfully. All available test cases passed.

Test case	Input (stdin)	Your Output (stdout)	Expected Output
Test case 0			
Test case 1	1 4 2 ball	1 4 2 ball	1 4 2 ball
Test case 2	3 all	3 all	3 all
Test case 3	4 call	4 call	4 call
Test case 4	5 bal	5 bal	5 bal
Test case 5			
Test case 6			

1 4  
2 ball  
3 all  
4 call  
5 bal  
1 3  
@ 一亩三分地

题目是看两两字符串构成的pair是否可以rearranged为回文串

既然可以rearrange那就简单很多

因为回文我们只care出现奇数次的char偶数次都会构成回文所以利用异或的性质( $x \wedge x = 0$ )来快速求解

要用到位运算所以可能不太好想到伪代码：

```
for word in words:  
    #字符串转换为integer便于位运算  
    bitmask=0  
    for c in word:  
        bitmask^=(1<<(ord(c)-ord('a')))  
    #checkbitmask是否在hash_map (key:bitmask value:count)中  
    #对于26位来说各异或一次并判断是否在hash_map中  
    #如果在就加上对应count的数目到result中
```

时间复杂度 $O(27n) \Rightarrow O(n)$

这道题应该是在说

```
Input: words = ["abcd", "dcab", "lls", "s", "sssll"]  
eg. abcd和dcab可以合成回文串, dcab可以变成dcba所以他们可以。
```

返回总共的数量。

```
public static int palindromePairs(String[] words) {  
    HashMap<Integer, Integer> hashmap = new HashMap<>();  
    int result = 0;  
    for (String word : words) {  
        int bitmask = 0;  
        for (int j = 0; j < word.length(); j++) {  
            bitmask ^= (1 << word.charAt(j) - 'a');  
        }  
        result += hashmap.getOrDefault(bitmask, 0);  
        hashmap.put(bitmask, hashmap.getOrDefault(bitmask, 0)+1);  
        for (int i = 0; i < 27; i++) {  
            int tmp = bitmask;  
            tmp ^= (1 << i);  
            result+= hashmap.getOrDefault(tmp, 0);  
        }  
    }  
    return result;  
}
```

## Question 2

Given a list of unique words, return the number of complementary pairs in the given list, so that the concatenation of the two words `words[i] + words[j]` is a palindrome.

### Example 1:

```
Input: words = ["abcd", "dcba", "lls", "s", "sssll"]
```

```
Output: 4
```

```
Explanation: The 4 palindromes are ["dcbaabcd", "abcddcba", "slls", "llssssll"]
```

### Example 2:

```
Input: words = ["bat", "tab", "cat"]
```

```
Output: 2
```

```
Explanation: The 2 palindromes are ["battab", "tabbat"]
```

## Function Description

Complete the function

`countComplementaiyPairs` in the editor below.

`countComplementaiyPairs` has the following parameters:

string `stringData[n]`, the strings to pair

## Returns

`longint`: the number of complementary pairs that can be formed

## Constraints

- $1 \leq n \leq 10^5$
- $1 \leq \text{length}(\text{stringData}[i]) \leq 3 \times 10^5$
- $1 \leq \text{sum of the length of strings in stringData} \leq 3 \times 10^5$
- All strings consist of lowercase English characters only.

✓✓ This Question is Same as Question on leetcode 336. Palindrome Pairs

Approach 1 : Brute Force

Approach 2 : Trie

Approach 3 : Bit Masking

## Binary Game

The screenshot shows a web browser window with the URL <https://hackerrank.com/test/8g48m93lhoo/questions/e6a5roce42s>. The page title is "Binary Game". The browser's address bar and navigation buttons are visible at the top. Below the address bar, there are several social media and platform logos: UW Handshake, Career & Internship, LeetCode - The W..., GoinGlobal!, and LinkedIn.

## 7. Binary Game

HackerBit designed a game based on binary digits. You are required to find the answer to the following problem to win the game.

A binary string is a string consisting only of digits 0 and 1. A binary string is said to be *good* if:

- The digit 1 only appears in groups of size *one\_group*, if it appears at all. For example, for *one\_group* = 2, "011<sub>orange</sub>110110" satisfies this condition while "01101010" does not.
- The digit 0 only appears in groups of size *zero\_group*, if it appears at all. For example, for *zero\_group* = 3, "11" satisfies this condition while "101010" does not.

For example, for *one\_group* = 2, *zero\_group* = 1, the strings "011", "000" are *good* while the strings "001" and "111" are not.

You are given four integers *min\_length*, *max\_length*, *one\_group*, and *zero\_group*. Find the number of *good* binary strings such that their lengths are in the range [*min\_length*, *max\_length*]. As the answer can be large, compute it modulo  $(10^9 + 7)$ .

### Example

Consider *min\_length* = 1, *max\_length* = 3, *one\_group* = 2 and *zero\_group* = 1.

The *good* strings satisfying the criteria are: ["0", "00", "11", "011", "110", "000"]. Hence, the answer is 6.



@一亩三分地

```
public static int BinaryGame(int min_length, int max_length, int one_group, int zero_group) {
    int result = 0;
```

```

int[] dp = new int[max_length + 1];
dp[0] = 1;
for (int i = 1; i <= max_length; i++) {
    if (i >= one_group)
        dp[i] += dp[i - one_group];
    if (i >= zero_group)
        dp[i] += dp[i - zero_group];
}

for (int i = min_length; i <= max_length; i++) {
    result+=dp[i];
}
return result;
}

```

## Array Generator

**1. Array Generator**

ALL A professor in the Computer Science department of HackerLand College wants to generate an array. Given an array of integers of length  $n$ ,  $arr$ , and two integers  $l$  and  $r$ , find another array,  $brr$ , such that:

- $l \leq brr[i] \leq r$
- $brr[i] - arr[i] < brr[i+1] - arr[i+1]$ , for every  $l$  less than  $n - 1$ .
- $brr[i] \leq brr[i+1]$  for every  $l$  less than  $n - 1$ .

Among all such arrays, return the lexicographically smallest one. If there is not an array that satisfies the conditions, then return an array with the single element -1.

**Example**

5  $arr = [1, 2, 1, 2], l = 1, r = 10$

10 #  
11 # C  
12 #  
13 # T  
14 # T  
15 # 1  
16 # 2  
17 # 3  
18 #  
19 #  
20 def e  
21  
22 P  
23  
24 #  
25 > if \_\_

6

The array [1, 3, 3, 5] satisfies given conditions.

7

1. Each element belongs in the range [1,10].
2. Construct an array  $crr$  where  $crr[i] = brr[i] - arr[i]$ ,  
 $crr = [0,1,2,3]$  and it is increasing.
3. The array  $brr$  is non-decreasing.

$brr[i] - arr[i]$  for each element is [0, 1, 2, 3]

$$1 - 1 = 0$$

$$3 - 2 = 1$$

$$3 - 1 = 2$$

$$5 - 2 = 3$$

Note that all  $brr[i] \geq arr[i]$  as well. There is not a lexicographically smaller array than [1, 3, 3, 5] that satisfies the conditions.

#### Function Description

Complete the function `getSmallestArray` in the editor below.

`getSmallestArray` has the following parameter(s):  
`int arr[n]:` an array of integers

#### Test Results



@一亩三分地

```
public static int[] ArrayGenerator(int[] arr, int l, int r){  
    if(arr.length==0) return new int[]{-1};  
    int[] brr = new int[arr.length];  
    int diff = 0;  
    int lastItem = arr[0];  
    for (int i = 0; i < arr.length; i++) {  
        if (arr[i]<1||arr[i]>10) return new int[]{-1};  
        brr[i]=Math.max(arr[i]+diff,lastItem);  
        diff = brr[i]-arr[i]+1;  
        lastItem = brr[i];  
    }  
    return brr;  
}
```

## Maximum Score

The screenshot shows a mobile browser displaying a HackerRank question. The top navigation bar includes icons for back, forward, and home, along with the URL <https://www.hackerrank.com/test/8g48m93lhoo/questions/>. A timer on the left indicates "59m left". On the right, there's a language selection dropdown with options 1 through 28. The main content area has a dark background with white text. It features a "BETA" button and a "Switch theme" link. The title "1. Maximum Score" is displayed prominently. To the left of the text are three circular icons: a refresh symbol, an information symbol, and a number 1. The text describes a problem involving books in a library and arrays, followed by instructions for performing moves on an array.

59m left

BETA Can't read the text? Switch theme

### 1. Maximum Score

ALL

There are several books placed in the HackerStudy library. One of those books on mathematics described an interesting problem that a mathematician wishes to solve.

1 For an array  $arr$  of  $n$  integers, the mathematician can perform the following move on the array:

2 1. Choose an index  $i$  ( $0 \leq i < \text{length}(arr)$ ) and add  $arr[i]$  to their scores.

Language: 1 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

3

2. Discard either the left partition (i.e.  $arr[0, 1, \dots, i-1]$ ) or the right partition (i.e.  $arr[i+1, i+2, \dots, length(arr)-1]$ ). The partition discarded can be empty too. The selected partition then becomes the new value of  $arr$  and is used for subsequent operations.

4

5

Starting with an initial score of 0, the mathematician wishes to find the maximum achievable score after  $k$  moves.

6

### Example

Consider  $n = 6$ ,  $arr = [4, 6, -10, -1, 10, -20]$ , and  $k = 4$

One optimal way the mathematician can perform the moves is as follows.

29

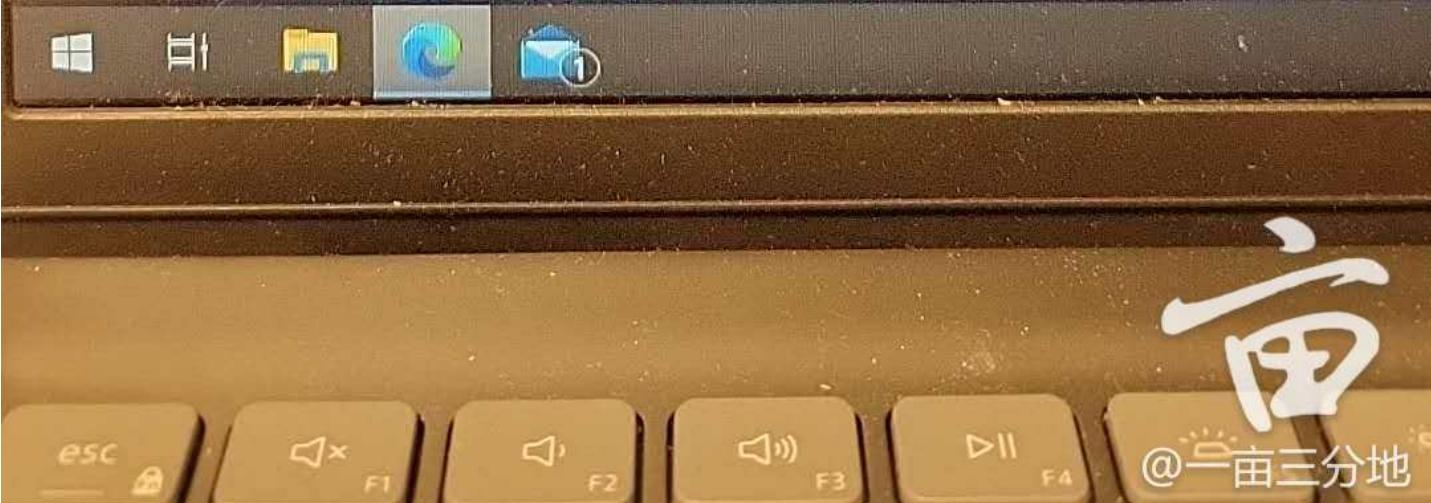
30

31

32

33 > p

Test Re



```
public static int MaximumScore(int n, int[] arr, int k){  
    if (k>n) k=n;  
    int result = 0;  
    PriorityQueue<Integer> pq = new PriorityQueue<>(Comparator.reverseOrder());  
    for (int j : arr) {  
        pq.add(j);  
    }  
    for (int i = 0; i < k; i++) {  
        if (!pq.isEmpty())  
            result += pq.poll();  
    }  
    return result;  
}
```

## Bank Transaction

R-76866 Software Development X HackerRank Question - Bank Tra X +  
https://www.hackerrank.com/test/8g48m93lhoo/questions/1mcblpst4

31m left



ALL



1



7

In a day, an account holder at HackerBank wants to make  $n$  transactions. In each transaction, money is either sent (negative amount) or received (positive amount). Given  $n$  transactions, the transactions occur in order from transaction 1 through transaction  $n$ , but transactions may be skipped. The balance starts at 0 and is the running sum of the selected transactions. It can never go negative.

Find out the maximum number of transactions possible.

#### Example

$transaction = [3, 2, -5, -6, -1, 4]$

One solution is to perform transactions 1, 2, 3, and 6. Transactions are  $0 + 3 + 2 + (-5) + 4$  and balances are  $[3, 5, 0, 4]$ . Return 4, the maximum number of transactions possible.

#### Function Description

Complete the function `maximizeTransactions` in the editor below.

`maximizeTransactions` has the following parameter(s):

`int transaction[n]:` the transaction amounts

#### Returns

`int:` the maximum number of transactions possible

Language Java 15

```
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51 public class
52     public st
53         Bu
54         Bu
55         ("OUTPU
56         nt t
57         @一亩三分地
Test Results
```

```
public static int BankTrans(int[] transaction) {
    int balance = 0;
    int result = 0;
```

```
PriorityQueue<Integer> pq = new PriorityQueue<>();
for (int trans : transaction) {
    if (trans >= 0) {
        balance += trans;
        result += 1;
    } else pq.add(trans);
}
while (!pq.isEmpty() && balance + pq.peek() >= 0) {
    balance += pq.poll();
    result += 1;
}
return result;

}
```

## Minimum Health

## 8. Minimum Health

Alex and *Charlie* are playing an online video game. Initially, there are  $m$  players in the first level, and there are next  $n$  levels. Each level introduces a new player (along with the players from the previous level). Each player has some strength which determines the difficulty of beating this player. To pass any level, select any available players and beat them.

Alex has completed the game and beaten the  $rank^{th}$  strongest player at every level. Now it's Charlie's turn to play. Whenever a player is beaten, Charlie's health decreases by the amount of strength of that player. So the initial health of Charlie must be greater than or equal to the sum of the strengths of players that are beaten throughout the game.

Charlie does not want to lose to Alex, so Charlie decided to also beat the  $rank^{th}$  strongest player at each level. What is the minimum initial health that Charlie needs to start with in order to do this?

### Example

`initial_players = [1, 2], new_players = [3, 4], rank = 2`

Charlie needs to beat the 2<sup>nd</sup> strongest player at each level.

For the first level, players have strengths 1 and 2, so Charlie needs to beat the player with strength 1.

For the second level, strengths are 1, 2, and 3, so Charlie defeats strength 2.

For the third level, strengths are 1, 2, 3, and 4, so Charlie defeats strength 3.

Total health needed =  $1 + 2 + 3 = 6$ .

### Function Description

Complete the function `getMinimumHealth` in the editor below.

`getMinimumHealth` has the following parameter(s):

`int initial_players[m]:` the strength of initial  $m$  players of the game

`int new_players[n]:` the strength of new  $n$  players that appear one by one after the first level

`int rank:` the rank that  $p2$  needs to beat at every level

### Returns

`long:` the initial health needed

### Constraints

- $1 \leq n, m \leq 10^5$
- $1 \leq rank \leq m$
- $1 \leq initial\_players[i], new\_players[i] \leq 10^9$



```
long result = 0;
int m = initial_players.length;
int n = new_players.length;
```

```

PriorityQueue<Integer> pq = new PriorityQueue<>();
for (int i = 0; i < rank; i++) {
    pq.add(initial_players[i]);
}
for (int i = rank; i < m; i++) {
    if(!pq.isEmpty() && pq.peek()<initial_players[i]){
        pq.poll();
        pq.add(initial_players[i]);
    }
}
if (!pq.isEmpty())
    result+= pq.peek();
System.out.println(result);
for (int new_player : new_players) {
    if (pq.peek() < new_player) {
        pq.poll();
        pq.add(new_player);
    }
    result += pq.peek();
}
return result;

```

## Finding integers

hackerrank.com/test/8g48m93lho/questions/295aq4mi8gp

AcWing Leetcode 一亩三分地 【新提醒】美国各... Resume Managem... LinkedIn Posting

54m left BETA Can't read the text? Switch theme

### 7. Finding Integers

ALL

There is a new assignment for the teacher's assessment evaluation for the students of HackerSchool. Start with a permutation of  $n$  integers from 1 through  $n$  stored in an array,  $arr$ , and an integer  $k$ . The  $i^{th}$  greatest element of an array is the  $i^{th}$  element of the array when sorted in decreasing order.

1 Find the  $k^{th}$  greatest element for each first  $i$  elements, where  $i$  ranges from  $[k, n]$ .

**Note:** A sequence of  $n$  integers is called a permutation if it contains every integer from 1 to  $n$  exactly once.

integers from 1 to n exactly once.

✓ **Example**

$n = 4, k = 2$

$arr = [4, 2, 1, 3]$



$i = 2, arr = [4, 2] \rightarrow$  The second greatest element is 2



$i = 3, arr = [4, 2, 1] \rightarrow$  The second greatest element is 2

$i = 4, arr = [4, 2, 1, 3] \rightarrow$  The second greatest element is 3



Return the array [2, 2, 3] as the answer.

7

**Function Description**

Complete the function *getGreatestElements* in the editor below.

*getGreatestElements* has the following parameter(s):

*int arr[n]:* a permutation array of  $n$  integers

*int k:* the rank of the element to find

**Returns**

*int[n - k + 1]:* answers for each  $i^{th}$  prefix in the range  $[k, n]$ .

**Constraints**

- $1 \leq n \leq 2 * 10^5$
- $1 \leq arr[i] \leq n$
- $1 \leq k \leq n$

Test Results

► Input Format for Custom Testing

一亩三分地  
@一亩三分地

```
public static int[] GetGreatestElements(int n, int k, int[] arr){  
    int[] res = new int[n-k+1];
```

```
int pos = 0;
PriorityQueue<Integer> pq = new PriorityQueue<>();
for (int i = 0; i < k; i++) {
    pq.add(arr[i]);
}
res[pos] = pq.peek();
pos++;

for (int i = k; i < n; i++) {
    if(!pq.isEmpty() && pq.peek()<arr[i]){
        pq.poll();
        pq.add(arr[i]);
    }
    res[pos] = pq.peek();
    pos++;
}
return res;
}
```

## Data updates

m left

BETA Can't read the text? [Switch theme](#)

Lang

✓ Auto

## 1. Data Updates

ALL

A data analyst recently joined HackerRank as an intern.

1

As an initial task, data for  $n$  days is provided to the intern. Then,  $k$  updates are performed on the data, where each update is of the form  $[l, r]$ . This indicates that the subarray of data starting at index  $l$  and ending at index  $r$  is negated. For example, if  $data = [1, 2, 3, 4]$  and the updates are  $[2, 4]$  then the data becomes  $data = [1, -2, -3, -4]$ .

2

Given the initial data and  $k$  updates, find the final data after all updates.

Note: 1-based indexing is used.

3

### Example

Consider  $n = 4$ ,  $data = [1, -4, -5, 2]$ ,  $k = 2$  and  $updates = [[2, 4], [1, 2]]$ .

4

1. After the first update, the data becomes  $data = [1, 4, 5, -2]$ .

2. After the second update, the data becomes  $data = [-1, -4, 5, -2]$ .

5

The final data is  $[-1, -4, 5, -2]$ .

6

### Function Description

Complete the function `getFinalData` in the editor below.

7

`getFinalData` has the following parameters:



@一亩三分地

```
public static int[] GetFinalData(int n, int[] data, int k, int[][] updates){  
    boolean[] booleans = new boolean[n];  
    for(int[] update: updates){  
        for (int i = update[0]-1; i <=update[1]-1 ; i++) {  
            booleans[i] = !booleans[i];  
        }  
    }  
    for (int i = 0; i < n; i++) {  
        if (booleans[i]){  
            data[i] *= -1;  
        }  
    }  
    return data;  
}
```

## Even Tag

One of the shops in HackerMall is offering discount coupons based on a puzzling problem. There are  $n$  tags where each tag has a value denoted by  $\text{val}[i]$ . A customer needs to choose the tags in such a way that the sum of values is even.

The goal is to find the maximum possible even sum of values of tags that can be chosen.

### Note:

- It is guaranteed that there is at least one tag with an even value.
- The tags can have positive or negative values.
- It can be possible to choose no tags at all.

### Example

Tag values are  $\text{val} = [2, 3, 6, -5, 10, 1, 1]$ .

Consider some of the following chosen tags and their corresponding sums:

Chosen Tags	Sum	Is sum even?
[2, 3, 6, 10, 1, 1]	23	No
[2, 3, 6, -5, 10, 1, 1]	18	Yes
[2, 6, 10, 1, 1]	20	Yes
[2, 3, 6, 10, 1]	22	Yes

The tags [2, 3, 6, 10, 1] sum to 22 which is even and is the maximum possible. Hence, the answer is 22.

@一亩三分地

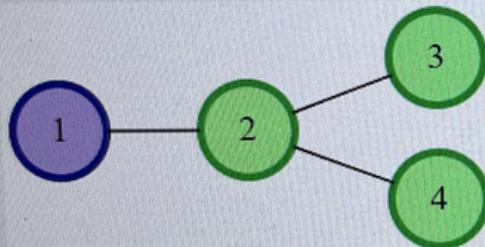
```
public static int EvenTag(int[] val){  
    int sum = 0;  
    int smallestOdd = Integer.MAX_VALUE;  
    for (int j : val) {  
        if (j > 0) {  
            sum += j;  
  
        }  
        if (j % 2 != 0) {  
            smallestOdd = Math.min(Math.abs(j), Math.abs(smallestOdd));  
        }  
    }  
    if (sum%2!=0){  
        sum-=smallestOdd;  
    }  
    return sum;  
}
```

## Delivery Manage System

## 1. Delivery Management System

A manufacturing company is located in a certain city. Their goods need to be shipped to other cities that are connected with bidirectional roads, though some cities may not be accessible because roads don't connect to them. The order of deliveries is determined first by distance, then by priority. Given the number of cities, their connections via roads, and what city the manufacturing company is located in, determine the order of cities where the goods will be delivered.

For example, let's say that the number of cities is *cityNodes* = 4, where *cityFrom* = [1, 2, 2], *cityTo* = [2, 4], and *company* = 1. In other words, the manufacturing company is located in city 1, and the roads run between cities 1 and 2, cities 2 and 3, and cities 2 and 4, like so:



In this case, the cities would be visited based on the following logic:

- The closest city (or cities) is visited first. This is city 2, which is 1 unit from the manufacturing company.
- The next-closest city (or cities) is visited next. This is city 3 and city 4, which are both 2 units from the manufacturing company.
  - In this case, priority is then calculated, visiting the smaller-numbered city first (city 3) and continuing in ascending order (city 4).

Therefore, the order is [2, 3, 4], which is the answer you would return.

### Function Description

Complete the function *order* in the editor below.

## Function Description

Complete the function *order* in the editor below.

*order* has the following parameters:

int *cityNodes*: the number of cities

int *cityFrom[n]*: the first city node where there is a bidirectional edge

int *cityTo[n]*: the second city node where there is a bidirectional edge

int *company*: the node where the route starts

Returns:

int[]: the cities where the goods will be delivered in the order visited

## Constraints

- $2 \leq \text{cityNodes} \leq 10^5$
- $1 \leq n \leq \min((\text{cityNodes} \times (\text{cityNodes} - 1)) / [2], 10^5)$
- $1 \leq \text{cityFrom}[i], \text{cityTo}[i], \text{company} \leq n$
- $\text{cityFrom}[i] \neq \text{cityTo}[i]$

## ▼ Sample Case 0

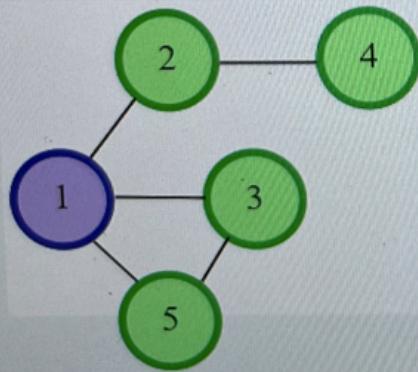
### Sample Input For Custom Testing

STDIN	Function
5 5	→ cityNodes = 5, n = 5
1 2	→ cityFrom = 1, cityTo = 2
1 3	→ cityFrom = 1, cityTo = 3
2 4	→ cityFrom = 2, cityTo = 4
3 5	→ cityFrom = 3, cityTo = 5
1 5	→ cityFrom = 1, cityTo = 5
1	→ company = 1

### Sample Output

2  
3  
5  
4

### Explanation



Cities 2, 3, and 5 are all 1 unit of distance away from the manufacturing company. These are visited based on priority in ascending order, so [2, 3, 5]. City 4 is 2 units of distance away from the manufacturing company, so it is visited next. Therefore, the final order is [2, 3, 5, 4].