1. Rohan is a B.tech student and he wants to make some games for his younger siblings where we want them to find the longest substring without repeating characters. Write java program for the same

Definition of Substring: A substring is a subset or part of another string, or it is a contiguous sequence of characters within a string.

Example 1:

Input: String => “abcabcabcbb”

Output: 3

Explanation: The answer is “abc” , with length of 3.

Example 2:

Input: String => “bbbbbbb”

Output: 1

Explanation: The answer is “b”, with length of 1.

Example 3:

Input: String => “xyxzxwzxy”

Output: 4

Explanation: The answer is “wzxy”, with length of 4.

Constraints:

1. 0 < string.length <=5\*10^4
2. String consists of English letters, digits, symbols and spaces.

2.John is a Computer Science teacher, how also happens to teach Mathematics part time to earn some extra bucks . He is teaching his students about Sets and subsets, Some student asks him about a question , help him to solve that question by providing a appropriate Java code.

Ques : There is an array of integers having unique elements, Provide all the possible subsets(The power sets).

The solution set must not contain duplicate subsets, but we free to return the solution in any order.

Example 1:

Input : Array => [1,2,3]

Output: [ [ ] ,[1] ,[2] ,[3] ,[1,2] ,[1,3] ,[2,3] ,[1,2,3] ]

Example 2:

Input: Array=> [0]

Output: [ [ ] , [0] ]

Constraints:

1. 1<=Array.length <=10
2. -10 <= Array[i] <= 10
3. All the elements of array are unique.

3.In semester exam students get there marks and teacher put those marks in a integer array of size K, now a student Smith wants to know his rank , So now teacher have to return the kth largest element in the array. We have to keep some things in mind that Kth largest element in the sorted order, not the kth distinct element.

Example 1:

Input : Marks\_List=> [90,94,20,95,80,78,69]

Smiths marks = 69

Output : Ranks = 6

Example 2:

Input : : Marks\_List=> [3,2,3,1,2,4,5,5,6]

Smiths marks = 4

Output : Ranks = 4

Constraints :

1. 1<= Smith’s Marks <=Marks\_List.length <=10^5
2. -10^4 <=Marks\_list[i] <=10^4

4.Rohan is a Stocks Market enthusiast and he likes to buy and sell penny shares and securities. He has an array of prices of shares where prices[i] is the price of given stocks on the (ith) day. He wants to maximize his profit by choosing a single day to buy one stock and chose a different day to sell the stock. Now you have to help him by writing a java code so that he can achieve the maximum profits from his transactions. If he cannot achieve any profits, then he returns 0.

Example 1:

Input: prices = [7,1,5,3,6,4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Example 2:

Input: prices = [7,6,4,3,1]

Output: 0

Explanation: In this case, no transactions are done and the max profit = 0.

Constraints:

1. 1 <= prices.length <= 10^5
2. 0 <= prices[i] <= 10^4

5.John is a student and has recently learned about Linked Lists and tried to solve a question in CodeChef about merging the sorted linked lists but unable to solve , but being a good student and his best friend you tried to help him.

Ques => There is an array of N linked-lists, each linked list is already sorted in Ascending order. Now you have to merge all of them and provide a new linked list in sorted form.

Example 1:

Input: lists = [[1,4,5],[1,3,4],[2,6]]

Output: [1,1,2,3,4,4,5,6]

Explanation: The linked-lists are:

[

1->4->5,

1->3->4,

2->6

]

merging them into one sorted list:

1->1->2->3->4->4->5->6

Example 2:

Input: lists = []

Output: []

Example 3:

Input: lists = [[]]

Output: []

Constraints:

1. k == lists.length
2. 0 <= k <= 104
3. 0 <= lists[i].length <= 500
4. -10^4 <= lists[i][j] <= 10^4
5. lists[i] is sorted in ascending order.
6. The sum of lists[i].length will not exceed 10^4.

6. Pooja is a Mathematics teachers and wants her students to find the power of various numbers, pow(x, n) which calculates x raised to the power n (i.e., xn). Now being a computer geek you wants to impress her and want to write a java program to solve this question.

Example 1:

Input: x = 2.00000, n = 10

Output: 1024.00000 Example 2:

Input: x = 2.10000, n = 3

Output: 9.26100 Example 3:

Input: x = 2.00000, n = -2

Output: 0.25000

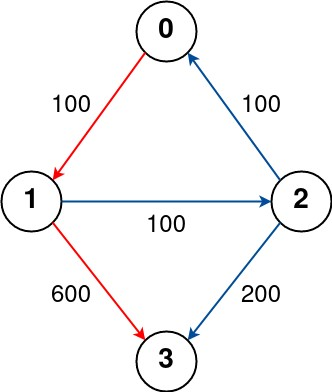
Explanation: 2-2 = 1/22 = 1/4 = 0.25

Constraints:

1. -100.0 < x < 100.0
2. -231 <= n <= 231-1
3. n is an integer.
4. -104 <= xn <= 104

7.John is a solo traveler and a vlogger and he likes to travel various places, but this time he lost his maps, mobile and wallet. Now he is only left with very less amount of money. He asked people for money and direction but many refused but, in the end, an old lady agrees to give him money if solves this question.

.



There are n cities connected by some number of flights. She also provides an array flights where flights[i] = [fromi, toi, pricei] indicates that there is a flight from city from[i] to city to[i] with cost price[i].

She also gives him knowledge about three integers Source City =src,

Destination City =dst, and No. of stops = k,

Now what is the minimum amount of money the old women will give John to travel between Source city and Destination city with at most K no. of stops If there is no such route, return -1.

Example 1:

Input: n = 4, flights = [[0,1,100],[1,2,100],[2,0,100],[1,3,600],[2,3,200]], src = 0, dst = 3, k = 1

Output: 700

Explanation:

The graph is shown above.

The optimal path with at most 1 stop from city 0 to 3 is marked in red and has cost 100 + 600 = 700.

Note that the path through cities [0,1,2,3] is cheaper but is invalid because it uses 2 stops.

Example 2:

Input: n = 3, flights = [[0,1,100],[1,2,100],[0,2,500]], src = 0, dst = 2, k = 1

Output: 200

Explanation:

The graph is shown above.

The optimal path with at most 1 stop from city 0 to 2 is marked in red and has cost 100 + 100 = 200.

Example 3:

Input: n = 3, flights = [[0,1,100],[1,2,100],[0,2,500]], src = 0, dst = 2, k = 0

Output: 500

Explanation:

The graph is shown above.The optimal path with no stops from city 0 to 2 is marked in red and has cost 500.

Constraints:

1. 1 <= n <= 100
2. 0 <= flights.length <= (n \* (n - 1) / 2)
3. flights[i].length == 3
4. 0 <= from[i], to[i] < n
5. from[i] != to[i]
6. 1 <= price[i] <= 10^4
7. There will not be any multiple flights between two cities.
8. 0 <= src, dst, k < n
9. src != dst

8. Smith creates another game for his siblings where he wants to find the length of longest common prefix in an array of strings.

If there is no common prefix, return an empty string "".

Example 1:

Input: strs = ["flower","flow","flight"]

Output: "fl"

Example 2:

Input: strs = ["dog","racecar","car"]

Output: ""

Explanation: There is no common prefix among the input strings.

Constraints:

1. 1 <= strs.length <= 200
2. 0 <= strs[i].length <= 200
3. strs[i] consists of only lowercase English letters.

9.Adelina is a nature lover and she wants to harvest rain water and tried to create a pit to collect rain water.



She create a pit in such a way that a every black block represents wall in the pit where width of each block is 1. Now compute how much water can we collect if it rains.

Input: height = [0,1,0,2,1,0,1,3,2,1,2,1]

Output: 6

Explanation: The above elevation map (black section) is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

Example 2:

Input: height = [4,2,0,3,2,5]

Output: 9

Constraints:

1. n == height.length
2. 1 <= n <= 2 \* 10^4
3. 0 <= height[i] <= 10^5

10. Priyank is travelling in plane for the first time and lost his ticket while boarding , so he sit randomly on someone else seat. There are total x passengers and total of x seats. If Priyank is the first one to sit and after him the rest of passengers are allowed to sit then the rest of the passengers will:

1. Take their own seat if it is still available, and
2. Pick other seats randomly when they find their seat occupied

Return the probability that the last or Xth person gets his own seat.

Example 1:

Input: n = 1

Output: 1.00000

Explanation: The first person can only get the first seat.

Example 2:

Input: n = 2

Output: 0.50000

Explanation: The second person has a probability of 0.5 to get the second seat (when first person gets the first seat).

Constraints:

1 <= n <= 10^5

11. Create a function that returns which chapter is **nearest** to the page you're on. If two chapters are equidistant, return the chapter with the **higher** page number.

**Examples**

nearestChapter(new Chapter[] {

new Chapter("Chapter 1", 1),

new Chapter("Chapter 2", 15),

new Chapter("Chapter 3", 37)

}, 10) ➞ "Chapter 2"

nearestChapter(new Chapter[] {

new Chapter("New Beginnings", 1),

new Chapter("Strange Developments", 62),

new Chapter("The End?", 194),

new Chapter("The True Ending", 460)

}, 200) ➞ "The End?"

nearestChapter(new Chapter[] {

new Chapter("Chapter 1a", 1),

new Chapter("Chapter 1b", 5)

}, 3) ➞ "Chapter 1b"

12. A number is said to be **Harshad** if it's *exactly divisible* by the **sum** of its digits. Create a function that determines whether a number is a Harshad or not.

### Examples

isHarshad(75) ➞ false

// 7 + 5 = 12

// 75 is not exactly divisible by 12

isHarshad(171) ➞ true

// 1 + 7 + 1 = 9

// 9 exactly divides 171

isHarshad(481) ➞ true

isHarshad(89) ➞ false

isHarshad(516) ➞ true

isHarshad(200) ➞ true