Q1.

Statement	Given a set of N strings. Each string consists of only lowercase alphabets. Sort each string in increasing order of the frequency of characters. If the frequency of two characters is the same, the character coming later in lexicographical order should come first. Input format: First line of input will contain the value of N. The next line will contain N space-separated strings. Output format: Multiple strings in a single line separated by space. Sample input: 2 little kettle Sample output: iettll lkttee
Test Case Input 1	2 aabbccddee pqrpqrpqr
Test Case Output 1	eeddccbbaa rrrqqqppp
Test Case Input 2	3 zyxabc temple torque
Test Case Output 2	zyxcba tpmlee utrqoe
Test Case Input 3	5 jdsbhchs bxsahybasyg uawjxasknhxcb ysabgchbcdhy sabhcjshdbchdgv
Test Case Output 3	jdcbsshh xhgyyssbbaa wusnkjhcbxxaa sgdayyhhccbb vjgassddccbbhhh
Test Case Input 4	6 hjasbxbc aygbx uayikiakaiia iaiaiaiaiai hnchcbdbchdy sbccdbgcgdv
Test Case Output 4	xsjhcabb yxgba yukkiiiiaaaa aaaaaiiiiii ynddbbhhhccc vsggddbbccc
Test Case Input 5	5 jdsbdasdsahchs bxfsarwqsahybasyg uawjxqgfdsasknhxcb ysabgcttretrehbcdhy sabhctreutyuafdjshdbchdgv
Test Case Output 5	jcbhhaadddssss xwrqhgfyybbsssaaa wuqnkjhgfdcbxxssaa sgdayyrrhheeccbbttt yvrjgfeuuttssccbbaahhhddd

Q2.

Statement	You are given an array of prices of size N. Where prices[i] is the price of a given stock on the ith day. You want to maximise your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0
Input Format	Two lines of input. The first line contains integer N. Next line contain array elements in space separated format
Output Format	Print the maximum profit as integer.
Constraint	2 <= N <= 30 1 <= prices[i] <= 10^3
Sample Input 1	6 7 1 5 3 6 4
Sample Output 1	5
Explanation	The 2nd day is the best day to buy the stock and the 5th day is best to sell that stock. The highest profit made is 5 (= 6 - 1)
Test Case Input 1	5 7 6 4 3 1
Test Case Output 1	0
Test Case Input 2	7 1 2 3 4 5 6 7
Test Case Output 2	6
Test Case Input 3	5 20 100 2 1 90

Test Case Output 3	89
Test Case Input 4	5 20 100 2 1 5
Test Case Output 4	80
Test Case Input 5	9 50 100 5 10 15 80 2 25 60
Test Case Output 5	75

Q3.

Statement	There are n people that are split into some unknown number of groups. Each person is labeled with a unique ID from 0 to n - 1. You are given an integer array groupSizes, where groupSizes[i] is the size of the group that person i is in. For example, if groupSizes[1] = 3, then person 1 must be in a group of size 3. Return a list of groups such that each person i is in a group of size groupSizes[i]. Each person should appear in exactly one group, and every person must be in a group. If there are multiple answers, return any of them. It is guaranteed that there will be at least one valid solution for the given input.
Input Format	The input (groupSizes) consists of a sequence of integers separated by spaces in single line .
Output Format	The output consists of a list of lists (or nested arrays).
Constraint	groupSizes.length == n 1 <= n <= 500 1 <= groupSizes[i] <= n
Sample Input 1	groupSizes = [3,3,3,3,3,1,3]
Sample Output 1	[[5],[0,1,2],[3,4,6]]
Explanation	The first group is [5]. The size is 1, and groupSizes[5] = 1. The second group is [0,1,2]. The size is 3, and groupSizes[0] = groupSizes[1] =

	groupSizes[2] = 3. The third group is [3,4,6]. The size is 3, and groupSizes[3] = groupSizes[4] = groupSizes[6] = 3. Other possible solutions are [[2,1,6],[5],[0,4,3]] and [[5],[0,6,2],[4,3,1]].
Test Case Input 1	groupSizes = [2,1,3,3,3,2]
Test Case Output 1	[[1],[0,5],[2,3,4]]

Q4.

Statement	Given a binary array nums, return the maximum length of a contiguous subarray with an equal number of 0 and 1. Expected time complexity is O(n).
Input Format	First line contains the value of N. The second line contains N space separated integers.
Output Format	A single integer.
Constraint	
Sample Input 1	3 0 1 0
Sample Output 1	2
Explanation	[0, 1] or [1, 0] is a longest contiguous subarray with equal number of 0 and 1.
Test Case Input 1	7 1 0 1 0 1 0 1
Test Case Output 1	6
Test Case Input 2	5 0 0 0 0 0
Test Case Output 2	0
Test Case Input 3	10 0 0 0 0 0 1 1 1 1 1
Test Case Output 3	10
Test Case Input 4	8 1 1 1 0 1 0 1 1
Test Case Output 4	4

Q5.

Statement	Geek just learned about Fibonacci numbers. The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, ... where the next number is found by adding up the two numbers before it. He defines a new series called Geeky numbers. Here the next number is the sum of the K preceding numbers. You are given an array of size K, GeekNum[ ], where the ith element of the array represents the ith Geeky number. Return its Nth term. **Note:** This problem can be solved in $O(N^2)$ time complexity but the user has to solve this in O(N). The Constraints are less because there can be integer overflow in the terms.
Input Format	Two lines of input. First line contains two space separated integer. Second line contains K space separated integers.
Output Format	Single integer in single line.
Constraint	$1 \leq K < 30$ $1 \leq N \leq 70$ $K \leq N$ $0 < GeekNum[ ] < 100$
Sample Input 1	5 3 0 1 2
Sample Output 1	6
Explanation	Terms are 0, 1, 2, 3, 6. So the 5th term is 6
Test Case Input 1	6 1 4
Test Case Output 1	4