4.Reverse Array

Write a program that takes an array of size N as input and gives the output as an array in the reverse order. The format of the input is as follow: N a1 a2 a3 a4 a5 … an 'N' is the size of the array and a1, a2, a3, … an, are its elements. Your program should give output (on the same line and separated by a space) as follows: an an-1 an-2 … a1

Example

Case 1: For input provided as follows: 3 1 2 3 Output of the program will be: 3 2 1 Description: As the input is 1 2 3, if we reverse these numbers we get: 3 2 1.

Case 2: For the input provided as follows: 4 1 2 1 0 Output of the program will be: 0 1 2 1 Description: The reversed order of the input is 0 1 2 1.

5.Palindrome

*Write a program that takes a string as input and prints whether this string is a true palindrome or not. A true palindrome is a string such that if you remove all characters except for letters (uppercase and lowercase), the string is equal if read from left to right and from right to left.*

*If the inputted string is a true palindrome, the message YES (uppercase required) is displayed else NO is displayed. Notice that lowercase letters are equal to uppercase letters and vice-versa.*

*Example*

*Case 1: For input provided as follows: abut-1-tuba Output of the program will be: YES Description: If we remove the bad characters 1 and -, we get the string “abuttuba” which is a palindrome.*

*Case 2: For the input provided as follows: @allula Output of the program will be: NO Description: After removing the bad characters we get “allula” which is not a palindrome.*

6.Permutation

*Write a program that takes two 11 integer array elements (one input per line) and gives output YES, if there's a permutation of the first array that is equal to the second array or gives output NO, if there's no such permutation. Example Case 1 For input provided as follows: 1 2 5 3 7 0 7 3 5 2 1 7 3 1 2 5 0 5 2 1 3 7 Output of the program will be: YES Description: We can check that the first array can be changed to the following array: 7 3 1 2 5 0 5 2 1 3 7 So that makes the first array a permutation of the second array, giving answer YES. Case 2 For the input provided as follows: 1 2 3 4 5 6 7 8 9 10 11 6 5 4 3 2 0 11 10 9 8 7 Output of the program will be: NO Description: The first array can not be a permutation of the second array since the first array doesn't have the number 0 that is present on the second array.*

7.Existing powers of 2

*You will be given a variable list of unknown length, containing 32-bits unsigned integers. You are required to give as output, a comma separated list in the increasing order, showing those powers of 2 that appear at least once, in the powers of 2 decomposition of the integers from the list. If no powers of 2 are present, you should give as output "NA", without quotes. For example, if the list is 1, 3, 4, you are expected to give as output 1, 2 and 4. Integers in the list will be entered one per line. Case 1: For the input provided as follows: 1 3 4 Output of the program will be: 1, 2, 4 Case 2: For the input provided as follows: 3 1 2 Output of the program will be: 1, 2*

8.Finding primes

*Given are two integers m, n (1 <= m <= n <= 10^9, n – m <= 10^5). You have to report the number of prime numbers within the interval [m, n]. First line of the input contains an integer T, denoting the test cases you have to handle. T lines follow, each one with two integers m, n as stated above. For each test case, output a single line, containing the requested report. Case 1: For the input provided as follows: 1 1 10 Output of the program will be: 4 Case 2: For the input provided as follows: 2 100 200 1 1 Output of the program will be: 21 0*

9.Optional Power

*Write a function in JS optionalPow that will receive two numeric arguments. The function is required to create some confirmation logic, where if the user agrees with the question (presses OK), then the function is required to compute the result of raising the first argument to the power determined by the second argument. If the user does not agree with the question (presses Cancel), then it should compute the power of the second argument raised to the first one. In both cases, the function must return the computed value. For example, optionalPow(2, 3) should return 8 if OK is pressed, otherwise it should return 9. The solution should be written in the following format: function optionalPow(a, b) { //your code here }*

10.Fix age

*Write a function in JS fixage that receives an array of integers. The integers represent ages between 0 and 100. The function is required to join all the integers that are between 18 and 60 into a string separated with a comma and returns it. If there is no integer between 18 and 60, then the function returns "NA". For example: fixage([5,15,25,78,59,45]) returns 25,59,45 and fixage([18,3,30,22,11,60]) returns 18,30,22,60 and fixage([1,3,3,2,11,6]) returns NA*

11.Common Digit

*Write a program that takes an integer N (N between 2 and 20 inclusive), and an array of N integers as input. Your program should output the most occurring digit that appears in the numbers of this array. If more than one digit satisfies this condition, output the bigger one. Note: The integer N will be given in the first line, and the array of N integers will be given in the second line. Each integer will be separated by a single space character ' '. Note2: A digit is an integer between 0 and 9. Example: Case 1: For input given as: 4 101 20 21 3 The output of the program will be: 1 Description of the output: The digit 0 appears 2 times (One time on the first number and another time on the second number), the digit 1 appears 3 times (Two times on the first number and another time on the third number), the digit 2 appears 2 times (One time on the second number and another time on the third number), and the digit 3 appears only one time, that is, on the last number. So the most common digit is 1. Case 2: For input given as: 3 111 222 5 The output of the program will be: 2 Description of the output: Both digits 1 and 2 appear the maximum number of times, that is, 3 times. Since 2 is bigger than 1, the answer is 2 instead of 1.*

12.Digit Sum

*Write a program that takes an integer N and an array of N positive integers as input. Your program should output the position (0-based) of the array in which the value in that position has the maximum digit sum possible. A digit sum of some value is the sum of its digits. If there's more than one position that satisfies this condition, print the greatest one. Note: The integer N will be given in the first line, and the array of N integers will be given in the second line (Each integer separated by a single space). Note1: 0-based position means that the first position of the array is equal to 0, the second is equal to 1, and so on. Example: Case 1: For input given as: 3 2 4 3 The output of the program will be: 1 Description of the output: The maximum digit sum we can get is 4 by getting the value in position 1 of the array. So, our answer is 1. Case 2: For input given as: 3 20 21 19 The output of the program will be: 2 Description of the output: The maximum digit sum we can get is 10 by summing up 1 and 9 from the number 19 in position 2 of the array. So, our answer is 2.*