**Задачи**

В каждой задаче необходимо написать программу, получающую входные параметры с консоли. Результат работы программы также печатается на консоль. Разрешается написание вспомогательных функций/структур/классов. Постарайтесь для каждой задачи найти оптимальное алгоритмческое решение, а так же предусмотреть граничные и исключительные случаи.

1. Написать функцию, которая печатает цифры заданного числа N в обратном порядке

***Входные данные:*** int N  
***Выходные данные:*** напечатанная строка  
***Примеры:***

* 123 => “321”
* -123 => “-321”
* 210 => “12”

1. Написать программу, которая проверяет четна ли сумма цифр заданного числа N.

***Входные данные:*** int N

***Выходные данные:*** строка “true” или “false”.

1. Напишите калькулятор. На вход подаётся два целых числа и символ, означающий математическое действие: +-\*/. Вам нужно вывести целочисленное решение на экран.

***Входные данные:*** int a, char c, int b  
***Выходные данные:*** a c b  
***Примеры:***

* + 10 + 20 => 30
  + 100 - 10 => 90
  + 10 \* 10 => 100
  + 10 / 10 => 1

1. На вход даны две строки без пробелов. В строках строго символы a-z и A-Z Определите, является ли одна строка перестановкой другой.

***Входные данные:*** char str[], str2[]  
***Выходные данные:*** true, если строки являются перестановкой, иначе false  
***Примеры:***

* + abc, cba => true
  + hello, ohlel => true
  + abc, cde => false

1. Дана строка, состоящая из символов (, ), {, }, [, ]. Необходимо определить, является ли данная строка валидной:

* Открывающаяся скобка должна закрываться скобкой этого же типа
* Скобки должны открываться и закрываться в правильном порядке

***Входные данные:*** char str[]  
***Выходные данные:*** true, если строка валидна, иначе false  
***Примеры:***

* () => true
* ()[]{} => true
* ({}) => true
* (] => false
* ([)] => false

1. Вася зарегистрирован в популярной социальной сети ВСкайнет. На сайте есть страница «Мои скайнетовцы», показывающая список друзей, упорядоченных по убыванию их популярности. Популярность друга определяется тем, как часто Вася заходил на его/ее страницу.

Вскайнет хранит историю посещений страниц друзей в виде массива чисел. Например, история [1, 1, 4, 3, 1, 4] означает, что Вася был на странице друга №1 три раза, два раза – на странице друга №4 и один раз – на странице друга №3.

По данной истории выведите на консоль номера друзей Васи в таком порядке, в каком их отображает ВСкайнет на странице «Мои скайнетовцы».

Если два друга одинаково популярны, то первым отображается тот, на чью страницу Вася заходил последним.

***Входные данные:***

* int friendsCount – количество друзей Васи;
* int N – длина массива с историей посещения страниц друзей.
* int[] history – массив с историей посещения страниц друзей. Гарантируется, что

;

***Выходные данные:*** номера друзей Васи