

**Обязательно ознакомьтесь с памяткой участника.**  
**Обратите внимание на следующие пункты:**

1. Используйте **только** ввод с клавиатуры и вывод на экран. Никакие файлы `input.txt/output.txt` или другие использовать нельзя.
2. Не используйте в Паскале модуль `crt` (и, следовательно, функцию `clrscr`).
3. Программа **должна заканчивать** работу после вывода выходных данных. Не используйте дополнительный `read/readln` или `system("pause")` в конце программы.
4. Не выводите на экран какую-либо лишнюю информацию, например: `writeln("Vvedite n: ")`. Программа должна выводить на экран **только** то, что написано в секции **Формат выходных данных** условия задачи.
5. Не проверяйте входные данные на корректность. Жюри **гарантирует**, что они удовлетворяют всем требованиям из секции задачи **Формат входных данных** условия задачи.

## **Памятка участника олимпиады**

Следуйте следующим правилам во время участия в олимпиаде:

1. Соревнование продлится **5 часов**.
2. Запрещается использовать любые электронные источники и носители информации, включая, но не ограничиваясь ими: портативные компьютеры, калькуляторы, радиотелефоны, пейджеры, дискеты, компакт-диски и прочее.
3. Запрещается общаться с другими участниками в любой форме.
4. Запрещается пользоваться Интернетом (за исключением доступа к тестирующей системе, см. пункт 13).
5. Решением является программа, написанная на одном из следующих языков программирования:
  - Pascal ABC
  - Free Pascal
  - C++ (Visual Studio 2010)
  - C++ 14 (MinGW 6)
  - Java 8
  - Python 3
  - C# (Visual Studio 2010, .NET 4.0)
6. На проверку сдается исходный текст программы, а не исполняемый файл. Программа должна полностью содержаться в одном файле, использование своих модулей и заголовочных файлов не допускается.
7. Программа должна читать входные данные только с клавиатуры и выводить результат на экран. Работать с файлами или какими-либо другими ресурсами компьютера запрещено.
8. Результаты работы программы проверяются автоматически, поэтому программа должна *точно соблюдать* формат вывода, указанный в условии. Не допускается вывод дополнительных сообщений. Ваша программа должна заканчивать свою работу после вывода ответа, а не «зависать» в ожидании ввода с клавиатуры.

9. Гарантируется, что входные данные будут соответствовать формату, указанному в условии. Все ограничения, заявленные в условии задачи, будут выполняться в тестах жюри. Проверять их в программе не имеет смысла. Ограничения по времени указаны для компьютера с тактовой частотой 3.2 ГГц.
10. Максимальный балл за каждую задачу указан в условии. Частичные решения будут оцениваться исходя из набора пройденных тестов.
11. Старайтесь использовать как минимум 32-х битные типы данных, если это позволяет ограничение по памяти (например, `longint` в Паскале).
12. Имя главного класса для программ на Java — `Solution` (см. пример программы на Java ниже).
13. Вы должны самостоятельно отсылать решения в тестирующую систему, используя сайт <http://acm.sgu.ru/contest-saratov>. Логин и пароль для входа на сайт вы должны получить от преподавателя. Обратите внимание, что пароль на второй тур новый. Ознакомьтесь с инструкцией, размещенной на сайте.
14. Результаты сразу за оба тура будут доступны на сайте <http://acm.sgu.ru/olimp> 23-го декабря, и также будут распространены через вашего учителя информатики. Удачи!

## Примеры решения задачи A+B

Ниже приведены решения задачи, в которой требуется считать два числа и вывести их сумму.

- Pascal:

```
var
  a, b: longint;
begin
  read(a, b);
  writeln(a + b);
end.
```

- C++:

```
#include <iostream>

using namespace std;

int main() {
  int a, b;
  cin >> a >> b;
  cout << a + b << endl;
  return 0;
}
```

- Java:

```
import java.util.Scanner;
import java.io.PrintWriter;

public class Solution {
  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    PrintWriter writer = new PrintWriter(System.out);
    int a = scan.nextInt();
    int b = scan.nextInt();
    writer.print(a + b);
    writer.close();
  }
}
```

- Python 3:

```
(a,b)=input().split()  
print(int(a) + int(b))
```

- C#:

```
using System;  
class Solution {  
    static void Main() {  
        String[] input = Console.ReadLine().Split(' ');  
        Console.WriteLine(Convert.ToInt32(input[0]) +  
            Convert.ToInt32(input[1]));  
    }  
}
```

## Задача 9. Новый пароль (15 баллов)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Поликарп решил придумать себе новый пароль, который удовлетворяет следующим условиям:

- длина пароля должна быть равна  $n$ ;
- пароль состоит только из цифр, отличных от нуля;
- каждая цифра в пароле должна встречаться не более одного раза;
- если представить запись пароля как десятичное число, то это число должно быть **нечетным**.

Среди всех подходящих паролей Поликарп выберет тот, десятичная запись которого является максимальным числом среди десятичных записей остальных паролей.

Перед вами стоит задача определить новый пароль Поликарпа, если вам задана длина пароля  $n$ .

### Формат входных данных

В первой строке следует целое число  $n$  ( $1 \leq n \leq 9$ ) — длина нового пароля Поликарпа.

### Формат выходных данных

Выведите одно целое число, состоящее из  $n$  цифр, — новый пароль Поликарпа.

### Примеры

стандартный ввод	стандартный вывод
1	9
2	97

## Задача 10. Путешествие (20 баллов)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вы прибыли в город, который расположен вдоль координатной оси  $OX$ .

В этом городе есть  $n$  достопримечательностей, которые вы бы хотели посетить, причем  $i$ -я достопримечательность имеет координату  $x_i$ . Вы готовились к путешествию заранее, поэтому пронумеровали все достопримечательности целыми числами от 1 до  $n$ , и для каждой пары различных достопримечательностей с номерами  $i$  и  $j$  верно, что если  $i < j$ , то  $x_i < x_j$ .

Изначально вы находитесь в координате, которая совпадает с координатой достопримечательности номер 1. Вы хотите посетить все достопримечательности, причем порядок, в котором вы будете посещать достопримечательности, не важен.

Перемещаться вы можете двумя способами:

- переместиться пешком на единицу расстояния в любом направлении (то есть влево или вправо); в этом случае уровень вашей усталости увеличится на  $y$ ;
- переместиться с помощью телепорта в любую точку города; в этом случае уровень вашей усталости увеличится на  $z$ .

Перед вами стоит задача определить минимальный уровень усталости, который может быть у вас после посещения всех достопримечательностей в произвольном порядке. Изначально ваш уровень усталости равен 0.

### Формат входных данных

В первой строке следуют три целых числа  $n, y, z$  ( $2 \leq n \leq 100, 1 \leq y, z \leq 100$ ).

Во второй строке следует последовательность различных целых чисел  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq 1000$ ) — координаты достопримечательностей в порядке слева направо. Гарантируется, что все элементы последовательности заданы в порядке строгого возрастания.

### Формат выходных данных

Выведите минимальный уровень усталости, который может быть у вас после посещения всех достопримечательностей в произвольном порядке.

### Примеры

стандартный ввод	стандартный вывод
4 2 5 5 6 9 11	11
8 1 70 6 13 24 25 30 51 67 70	64
7 1 2 14 25 30 58 62 89 93	12

### Замечание

В первом примере можно, например, переместиться пешком от первой достопримечательности до второй на расстояние 1, затем телепортироваться к третьей достопримечательности, а после этого переместиться пешком до четвертой достопримечательности на расстояние 2. В этом случае уровень усталости будет равен  $2 \cdot 1 + 5 + 2 \cdot 2 = 11$ .

Во втором примере не нужно телепортироваться, а нужно просто переместиться пешком от первой достопримечательности до последней.

## Задача 11. Одинаковые буквы (30 баллов)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

У вас есть строка  $s$  длины  $n$ , состоящая из строчных букв латинского алфавита. К каждой из позиций строки вы можете применить следующую операцию **не более одного раза**: заменить букву в позиции на одну из соседних букв в алфавите.

Например, букву «n» можно заменить на букву «m» или на букву «o». Так как буква «a» является первой в алфавите, то ее можно заменить только на букву «b», а так как буква «z» — последняя буква в алфавите, то ее можно заменить только на букву «y».

Определите минимальное количество операций, которые нужно произвести с заданной строкой  $s$  таким образом, чтобы все буквы в строке  $s$  стали одинаковыми. Если это невозможно, сообщите об этом.

### Формат входных данных

В первой строке следует целое число  $n$  ( $2 \leq n \leq 200\,000$ ) — количество букв в строке  $s$ .

Во второй строке следует строка  $s$  длины  $n$ , состоящая из строчных букв латинского алфавита.

### Формат выходных данных

Если невозможно сделать все буквы в строке  $s$  одинаковыми, выведите  $-1$ .

В противном случае, выведите минимальное количество описанных в условии операций, которые нужно произвести со строкой  $s$  таким образом, чтобы все буквы в строке  $s$  стали одинаковыми.

### Примеры

стандартный ввод	стандартный вывод
5 abcab	3
3 ggg	0
10 abcdevwxyz	-1

### Замечание

В первом примере нужно заменить буквы «a» в первой и четвертой позициях на «b», а также заменить букву «c» в третьей позиции на «b». Тогда получится строка, целиком состоящая из букв «b». Таким образом, нам понадобится 3 операции.

Во втором примере не нужно совершать никаких операций, так как строка изначально состоит из одинаковых букв.

В третьем примере невозможно сделать так, чтобы строка состояла из одинаковых букв.

## Задача 12. Башни (35 баллов)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

У Толи есть кубики, из которых он построил  $n$  башен. Башни стоят в ряд и пронумерованы слева направо, начиная с единицы. Каждая башня — это некоторое положительное количество кубиков, которые поставлены друг на друга. Башня номер  $i$  имеет высоту  $a_i$ , то есть состоит из  $a_i$  кубиков.

Толя решил сделать из последовательности башен палиндромную последовательность. Он хочет сделать так, чтобы высота первой башни стала равна высоте последней башни, высота второй башни стала равна высоте предпоследней башни, и так далее. Иными словами, Толя хочет, чтобы высота башни номер  $i$  стала равна высоте башни номер  $(n - i + 1)$ .

Для достижения своей цели Толя может выполнять следующее действие: взять самый верхний кубик любой башни и положить взятый кубик наверх другой башни. Толя будет брать кубики только из таких башен, которые состоят хотя бы из двух кубиков.

Толя хочет сделать палиндромную последовательность как можно скорее, поэтому вам нужно помочь ему и определить минимальное количество действий, после которых последовательность башен Толи станет палиндромной. Если невозможно получить палиндромную последовательность, сообщите об этом.

### Формат входных данных

В первой строке следует целое число  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — количество башен.

Во второй строке следует последовательность целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ), где  $a_i$  равно высоте  $i$ -й башни.

### Формат выходных данных

Выведите минимальное количество действий, описанных в условии, которые нужно совершить, чтобы последовательность башен Толи стала палиндромной. Если невозможно получить палиндромную последовательность, выведите  $-1$ .

## Примеры

стандартный ввод	стандартный вывод
5 2 4 1 3 1	1
2 1 2	-1
8 2 1 8 5 6 3 4 1	5

## Замечание

В первом примере Толя может взять верхний кубик второй башни и положить его наверх пятой башни. Тогда последовательность высот башен станет равна  $[2, 3, 1, 3, 2]$ . Эта последовательность является палиндромной, поэтому Толе достаточно одного действия.

Во второй примере невозможно сделать из последовательности башен палиндромную последовательность.

В третьем примере одна из оптимальных последовательностей действий следующая:

1. взять верхний кубик третьей башни и положить его наверх четвертой башни, последовательность высот станет равна  $[2, 1, 7, 6, 6, 3, 4, 1]$ ;
2. взять верхний кубик первой башни и положить его наверх шестой башни, последовательность высот станет равна  $[1, 1, 7, 6, 6, 4, 4, 1]$ ;
3. взять верхний кубик третьей башни и положить его наверх второй башни, последовательность высот станет равна  $[1, 2, 6, 6, 6, 4, 4, 1]$ ;
4. взять верхний кубик третьей башни и положить его наверх второй башни, последовательность высот станет равна  $[1, 3, 5, 6, 6, 4, 4, 1]$ ;
5. взять верхний кубик третьей башни и положить его наверх второй башни, последовательность высот станет равна  $[1, 4, 4, 6, 6, 4, 4, 1]$ .

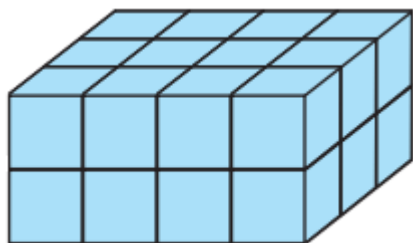
### Задача 13. Параллелепипед (40 баллов)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

У вас есть  $n$  одинаковых кубиков. Вы можете составлять из кубиков различные геометрические фигуры.

Перед вами стоит задача определить количество способов составить из имеющихся  $n$  кубиков прямоугольный параллелепипед. Использовать все кубики обязательно.

На рисунке изображен параллелепипед, который составлен из  $2 \cdot 3 \cdot 4 = 24$  кубиков.



Два параллелепипеда считаются одинаковыми, если один можно получить из другого с помощью поворотов.

#### Формат входных данных

В первой строке следует целое число  $n$  ( $1 \leq n \leq 10^{12}$ ) — количество кубиков.

Гарантируется, что не менее чем в половине тестов  $n$  не превосходит  $10^6$ .

#### Формат выходных данных

Выведите количество прямоугольных параллелепипедов, которые можно составить из  $n$  кубиков (использовать все кубики обязательно).

#### Примеры

стандартный ввод	стандартный вывод
6	2
1	1
18	4
1000000	139
1000000000000	1405

#### Замечание

В первом примере из шести кубиков можно составить 2 параллелепипеда с размерами  $1 \times 1 \times 6$  и  $1 \times 2 \times 3$ .

Во втором примере всего один кубик, который сам по себе является параллелепипедом  $1 \times 1 \times 1$ .

В третьем примере 18 кубиков, и из них можно составить 4 параллелепипеда с размерами  $1 \times 1 \times 18$ ,  $1 \times 2 \times 9$ ,  $1 \times 3 \times 6$  и  $2 \times 3 \times 3$ .

### Задача 14. Игра с деревом (45 баллов)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В этот раз Аня и Боря решили сыграть в игру, используя дерево из  $n$  вершин. Дерево из  $n$  вершин — это связный ациклический граф, состоящий из  $n - 1$  ребра.

Изначально, вершина 1 покрашена в черный цвет, а вершина  $n$  покрашена в белый цвет. Остальные вершины дерева не покрашены. Аня и Боря будут ходить по очереди, первый ход в игре сделает Аня. Во время своих ходов игроки должны сделать следующее:

- Аня во время своего хода должна выбрать непокрашенную вершину  $x$ , которая соединена ребром с вершиной, покрашенной в черный цвет, и покрасить вершину  $x$  в черный цвет;
- Боря во время своего хода должен выбрать непокрашенную вершину  $y$ , которая соединена ребром с вершиной, покрашенной в белый цвет, и покрасить вершину  $y$  в белый цвет.

Если игрок не может сделать ход, он проигрывает игру.

Определите победителя игры, если оба игрока будут действовать оптимально.

### Формат входных данных

В первой строке следует целое число  $n$  ( $2 \leq n \leq 100\,000$ ) — количество вершин в дереве.

В  $i$ -й из следующих  $(n - 1)$  строк следуют два целых числа  $v_i, u_i$  ( $1 \leq v_i, u_i \leq n, v_i \neq u_i$ ) — номера вершин, соединенных  $i$ -м ребром. Гарантируется, что заданные ребра образуют дерево.

### Формат выходных данных

Если игру выиграет Аня, выведите «Аня» (без кавычек).

Если игру выиграет Боря, выведите «Borya» (без кавычек).

### Примеры

стандартный ввод	стандартный вывод
4 4 3 1 2 2 3	Borya
7 2 1 3 1 4 1 4 7 6 3 5 7	Anya
4 2 4 4 1 3 2	Borya

### Замечание

В первом примере вершина 1 изначально черного цвета, а вершина 4 — белого. Аня во время своего первого хода должна покрасить вершину 2 в черный цвет. После этого будет ходить Боря, и он во время своего первого хода должен покрасить вершину 3 в белый цвет. После этого Аня не сможет сделать ход, поэтому Боря выиграет.

## Задача 15. Добавление чисел (55 баллов)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В данной задаче вам задано дерево, состоящее из  $n$  вершин. Известно, что ровно в  $k$  вершинах дерева записано по одному целому числу. Остальные  $(n - k)$  вершин пустые, то есть в них ничего не записано.

Перед вами стоит задача записать во все пустые вершины ровно по одному целому числу таким образом, что если две вершины соединены ребром, то модуль разности между числами, записанными в этих вершинах, должен быть равен 1. Если это невозможно, сообщите об этом.

Вы не можете изменять числа в тех  $k$  вершинах, в которых они изначально были записаны.

### Формат входных данных

В первой строке следует целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество вершин в дереве.

В  $i$ -й из следующих  $(n - 1)$  строк следуют два целых числа  $v_i, u_i$  ( $1 \leq v_i, u_i \leq n, v_i \neq u_i$ ) — номера вершин, соединенных  $i$ -м ребром. Гарантируется, что заданные ребра образуют дерево.

В следующей строке следует целое число  $k$  ( $1 \leq k \leq n$ ) — количество вершин дерева, в которых изначально записаны числа.

В  $j$ -й из следующих  $k$  строк следуют два целых числа  $a_i, p_i$  ( $1 \leq a_i \leq n, 0 \leq p_i \leq 1\,000\,000$ ), где  $p_i$  равно числу, изначально записанному в вершину  $a_i$ . Гарантируется, что все  $a_i$  различны.

### Формат выходных данных

Если невозможно выполнить описанное в условии, выведите «No» (без кавычек).

В противном случае, в первую строку выведите «Yes». Во вторую строку выведите  $n$  целых чисел, причем  $i$ -е число должно быть равно числу, записанному в  $i$ -й вершине. Числа в  $k$  вершинах, которые были изначально записаны в них, не должны быть изменены и должны быть равны числам из входных данных. Если ответов несколько, разрешается вывести любой из них. Обратите внимание, что ответные числа для вершин, которые изначально были пустыми, могут быть отрицательными или могут превосходить число  $10^6$ .

### Примеры

стандартный ввод	стандартный вывод
5 1 3 5 3 3 4 2 1 2 5 9 2 8	Yes 7 8 8 7 9
5 3 4 2 1 1 3 3 5 3 5 7 4 3 2 6	No
4 4 3 3 2 2 1 1 1 0	Yes 0 -1 -2 -3

### Замечание

В первом примере в первую вершину нужно записать число 7, в третью вершину – число 8, а в четвертую – число 7.

## Задача 16. Новая перестановка (60 баллов)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

У вас есть перестановка  $a$  длины  $n$ , причем  $n$  — четно.

Перестановка длины  $n$  — это последовательность, состоящая из  $n$  целых чисел, причем каждое из целых чисел от 1 до  $n$  встречается в этой последовательности ровно один раз.

Перед вами стоит задача построить перестановку  $b$  длины  $n$  по следующим правилам. Изначально,  $b$  — это пустая последовательность. Затем нужно производить следующие операции до тех пор, пока  $a$  не станет пустой (каждая операция состоит из трех этапов):

1. выбрать два **соседних** элемента в  $a$ , назовем их  $p$  и  $q$ ;
2. удалить  $p$  и  $q$  из  $a$  (таким образом, длина  $a$  уменьшится на два);
3. добавить  $p$  и  $q$  в **начало**  $b$ , сохранив их относительный порядок.

Легко показать, что после того, как  $a$  станет пустой, последовательность  $b$  будет являться перестановкой длины  $n$ .

Перед вами стоит задача определить лексикографически минимальную перестановку  $b$ , которую можно получить с помощью описанных операций.

### Формат входных данных

В первой строке следует целое число  $n$  ( $2 \leq n \leq 200\,000$ ) — длина перестановки  $a$ . Гарантируется, что  $n$  — четное число.

Во второй строке следует последовательность целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — перестановка  $a$ . Гарантируется, что заданная последовательность является перестановкой.

### Формат выходных данных

Выведите  $n$  целых чисел — лексикографически минимальную перестановку  $b$ , которую можно получить из перестановки  $a$  с помощью описанных в условии операций.

### Примеры

стандартный ввод	стандартный вывод
8 5 4 1 7 6 8 2 3	1 3 5 4 7 2 6 8
6 2 6 3 5 1 4	1 4 2 5 6 3
10 4 5 8 3 9 10 2 1 7 6	2 1 4 3 5 8 7 6 9 10



### Замечание

Рассмотрим подробнее первый пример (изначально,  $a = [5, 4, 1, 7, 6, 8, 2, 3]$ ,  $b = []$ ):

1. выберем элементы 6 и 8, удалим их из  $a$  и добавим их в начало  $b$ , после этого  $a = [5, 4, 1, 7, 2, 3]$ ,  $b = [6, 8]$ ;
2. выберем элементы 7 и 2, удалим их из  $a$  и добавим их в начало  $b$ , после этого  $a = [5, 4, 1, 3]$ ,  $b = [7, 2, 6, 8]$ ;
3. выберем элементы 5 и 4, удалим их из  $a$  и добавим их в начало  $b$ , после этого  $a = [1, 3]$ ,  $b = [5, 4, 7, 2, 6, 8]$ ;
4. выберем элементы 1 и 3, удалим их из  $a$  и добавим их в начало  $b$ , после этого  $a = []$ ,  $b = [1, 3, 5, 4, 7, 2, 6, 8]$ .