

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 **4 часа**.
2. Запрещается использовать любые электронные источники и носители информации, включая, но не ограничиваясь ими: портативные компьютеры, калькуляторы, радиотелефоны, пейджеры, дискеты, компакт-дискеты 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 — расширение `pas`
 - Free Pascal — расширение `pp`
 - C++ (Visual Studio 2010) — расширение `cpp`
 - C++ (MinGW 4.7) — расширение `sxx`
 - Java 7 — расширение `java`
 - Python 3 — расширение `py`
 - C# (Visual Studio 2010, .NET 4.0) — расширение `cs`
6. На проверку отдается 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 количества 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. Если вы участвуете через сайт <http://acm.sgu.ru/contest>, то вы самостоятельно отправляете решения в тестирующую систему. Логин и пароль для входа на сайт вы должны получить от преподавателя. Ознакомьтесь с инструкцией, размещенной на сайте.

14. В противном случае ваши решения будут собраны с вашего компьютера преподавателем по окончании работы. Для этого именуруйте файлы своих решений таким образом: `xxxxdd.ext`, где `xxxx` — это ваш четырехбуквенный идентификатор, `dd` — это номер задачи, а `ext` — это расширение вашего языка программирования. Например, имя файла для решения задачи 01 участника `abcd` на языке Pascal должно иметь вид `abcd01.pas`. Решения на Java следует именовать таким же образом, при компиляции вашей программы файл будет переименован в `Solution.java`.
15. Результаты будут доступны через сутки после окончания или ранее на сайте <http://acm.sgu.ru/olimp>, и также будут распространены через вашего учителя информатики.

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:

```
(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 01. Хипстер Вася (60 баллов)

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

Как-то раз хипстер Вася решил посчитать, сколько у него носков. Оказалось, что у Васи есть a красных носков и b синих носков.

Согласно последним веяниям моды, хипстеры должны носить носки разных цветов: на левой ноге красный, а на правой — синий.

Каждый день с утра Вася надевает новые носки, а перед сном выкидывает их, ведь стирать ему их очень не хочется.

Васе стало интересно, какое максимальное количество дней, в которые он сможет быть одетым по моде и носить разноцветные носки, и сколько затем дней он сможет ходить в одноцветных носках, пока носки либо не закончатся, либо из оставшихся носков нельзя будет составить ни одной пары.

Поможете ему?

Формат входных данных

В единственной строке входных данных следуют два целых положительных числа a и b ($1 \leq a, b \leq 100$) — количество носков красного и синего цветов, которые есть у Васи.

Формат выходных данных

Выведите через пробел два целых числа — максимальное количество дней, в которые Вася сможет носить разноцветные носки, и количество дней, которое он сможет носить одноцветные носки до тех пор, пока они либо не кончатся, либо из оставшихся носков нельзя будет составить ни одной пары.

Напомним, что в конце дня Вася выкидывает носки, которые он носил в этот день.

Примеры

стандартный ввод	стандартный вывод
3 1	1 1
2 3	2 0
7 3	3 2

Замечание

В первом примере Вася сначала может надеть одну пару разноцветных носков, после этого у него останется два красных носка, которые он наденет во второй день.

Задача 02. Элитные дома (60 баллов)

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

В столице Берляндии есть n многоэтажных домов. Архитектор, который застраивал столицу, был человеком очень

творческим, поэтому все дома были построены в один ряд.

Пронумеруем все дома слева направо, начиная с единицы. Дом считается *элитным*, если количество этажей в нем строго больше, чем во всех домах с большими номерами. Иными словами, дом элитный, если количество этажей в нем строго больше, чем во всех домах, которые расположены правее, чем он. В данной задаче полагается, что высоты всех этажей во всех домах одинаковы.

Нового архитектора заинтересовали n вопросов, i -й из которых звучит так: «сколько этажей нужно достроить к i -му дому, чтобы он стал элитным?» (где i принимает все значения от 1 до n). Вам предстоит помочь ему справиться с этой задачей.

Заметим, что все вопросы независимы друг от друга — ответ на вопрос для дома i не влияет на другие ответы (то есть, этажи к домам на самом деле не достраиваются).

Формат входных данных

В первой строке входных данных записано единственное целое число n ($1 \leq n \leq 100$) — количество домов столице Берляндии.

Во второй строке через пробел записаны n целых положительных чисел h_i ($1 \leq h_i \leq 100$), где h_i равно количеству этажей в i -м доме.

Формат выходных данных

Выведите n целых чисел a_1, a_2, \dots, a_n , где число a_i равно количеству этажей, которые необходимо достроить к дому номер i , чтобы он стал элитным. Если дом уже элитный и к нему достраивать ничего не нужно, тогда a_i должно быть равно нулю.

Все дома нумеруются слева направо, начиная с единицы.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 1 2	3 2 0 2 0
4 3 2 1 4	2 3 4 0

Задача 03. Школьники в библиотеке (60 баллов)

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

В конце учебного года n школьников пришли сдать учебники в библиотеку. Все они выстроились в очередь. Затем каждый школьник посчитал суммарное количество книг, которые были у него и всех тех, кто стоял в очереди раньше него.

Вам даны посчитанные числа в некотором порядке. Необходимо определить, сколько книг было у каждого из школьников.

Гарантируется, что у каждого школьника с собой была хотя бы одна книга.

Формат входных данных

В первой строке входных данных находится целое положительное число n ($1 \leq n \leq 100$) — количество школьников, которые пришли в библиотеку.

В следующей строке находится n различных целых положительных чисел a_i ($1 \leq a_i \leq 10000$), где число a_i равно количеству книг, которое посчитал некоторый школьник, стоявший в очереди.

Формат выходных данных

Выведите n целых положительных чисел — количество книг каждого из школьников. Первое число должно быть равно количеству книг, которое принес в библиотеку школьник, стоящий в очереди первым, затем число, которое равно количеству книг, которое принес с собой школьник, стоявший в очереди на втором месте, затем на третьем месте и так далее.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2	1 1 1
4 10 9 7 4	4 3 2 1

Замечание

В первом примере каждый из трех школьников в очереди держал одну книгу. Таким образом, первый из них посчитал ровно одну книгу, второй — две, а третий — три. Эти числа и заданы во входных данных в некотором порядке.

Задача 04. Три логотипа (60 баллов)

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

Три компании решили заказать рекламный щит с изображениями своих логотипов. Рекламный щит представляет собой большую **квадратную** доску. Логотип каждой из компаний представляет собой прямоугольник ненулевой площади.

Рекламодатели разместят рекламу только в том случае, если возможно расположить все три логотипа на щите без наложений так, что на нём не останется свободного места. При размещении логотипы можно поворачивать так, чтобы стороны были параллельны сторонам рекламного щита.

Перед вами стоит задача определить, возможно ли разместить логотипы трех компаний на щите квадратной формы, не нарушив никакого из описанных правил.

Формат входных данных

В единственной строке входных данных находятся шесть целых положительных чисел $x_1, y_1, x_2, y_2, x_3, y_3$ ($1 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 100$), где x_i и y_i обозначают длину и ширину логотипа i -й компании соответственно.

Формат выходных данных

Если разместить все три логотипа на квадратном щите невозможно, выведите одно число “-1” (без кавычек).

Если же это возможно, выведите в первую строку длину стороны квадрата n , в котором можно разместить все три логотипа. В следующих n строках должно содержаться по n заглавных латинских букв “А”, “В” или “С”. Наборы одинаковых букв должны образовывать сплошные прямоугольники, причём:

- размеры прямоугольника, составленного из букв “А”, должны быть равны размерам логотипа первой компании,
- размеры прямоугольника, составленного из букв “В”, должны быть равны размерам логотипа второй компании,
- размеры прямоугольника, составленного из букв “С”, должны быть равны размерам логотипа третьей компании.

Напомним, что логотипы компаний в процессе нанесения их на рекламный щит можно поворачивать. Пустых мест на щите остаться не должно. Если квадратный рекламный щит можно заполнить логотипами несколькими способами, разрешается вывести любой из них.

Изучите тестовые примеры для лучшего понимания условия.

Примеры

стандартный ввод	стандартный вывод
5 1 2 5 5 2	5 AAAAA BBBBB BBBBB CCCCC CCCCC
4 4 2 6 4 2	6 BBBBBB BBBBBB AAAACC AAAACC AAAACC AAAACC

Примеры

стандартный ввод	стандартный вывод
2 4 7 9	2
3 8 17 15 19	5
2 2 99 100	20

Замечание

В первом тестовом примере оптимальная стратегия следующая. Петя должен улучшить первый навык до 10, потратив 3 единицы улучшения, и второй навык до 10, потратив одну единицу улучшения. Таким образом, Петя потратит все свои единицы улучшения и суммарный рейтинг главного героя станет равен $\lfloor \frac{10}{10} \rfloor + \lfloor \frac{10}{10} \rfloor = 1 + 1 = 2$.

Во втором тестовом примере оптимально для Пети улучшить первый навык до 20 (при этом потратить 3 единицы улучшения) и улучшить третий навык до 20 (при этом потратить 1 единицу улучшения). Таким образом, у Пети останется 4 единицы улучшения и он сможет увеличить второй навык лишь до 19 (что не изменит суммарный рейтинг, поэтому делать это Пете необязательно). Поэтому максимально возможный суммарный рейтинг в этом примере $\lfloor \frac{20}{10} \rfloor + \lfloor \frac{19}{10} \rfloor + \lfloor \frac{20}{10} \rfloor = 2 + 1 + 2 = 5$.

В третьем тестовом примере оптимальная стратегия Пети — увеличить первый навык до 100, потратив 1 единицу улучшения. После этого оба навыка главного героя станут равны 100, поэтому Петя не сможет потратить оставшуюся единицу улучшения. Таким образом, ответ равен $\lfloor \frac{100}{10} \rfloor + \lfloor \frac{100}{10} \rfloor = 10 + 10 = 20$.

Задача 05. Развитие навыков (60 баллов)

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

Петя очень любит компьютерные игры. Наконец вышла игра, которую он так долго ждал!

У главного героя этой игры есть n различных навыков, каждый из которых характеризуется целым числом a_i от 0 до 100. Чем больше число a_i , тем лучше главный герой владеет i -м навыком. Суммарный рейтинг главного героя определяется как сумма значений $\lfloor \frac{a_i}{10} \rfloor$ по всем i от 1 до n . Выражение $\lfloor x \rfloor$ обозначает результат округления числа x вниз к ближайшему целому числу.

Пете в начале игры в качестве бонуса дали k единиц улучшений, которые он может использовать для увеличения навыков своего персонажа и его суммарного рейтинга. Используя одну единицу улучшения, Петя может увеличить любой навык главного героя ровно на один. К примеру, если $a_4 = 46$, то после применения единицы улучшения к этому навыку, он станет равен 47. Больше 100 навык героя стать не может. Таким образом, возможно, что некоторое количество единиц улучшения останется невостребованным.

Перед вами стоит задача определить, как следует использовать единицы улучшений, чтобы максимизировать суммарный рейтинг главного героя. Использовать все единицы улучшений не обязательно.

Формат входных данных

В первой строке входных данных следуют два целых положительных числа n и k ($1 \leq n \leq 10^5$, $0 \leq k \leq 10^7$) — количество навыков главного героя и количество единиц улучшений, имеющихся у Пети.

Во второй строке входных данных следует последовательность из n целых чисел a_i ($0 \leq a_i \leq 100$), где a_i характеризует уровень развитости i -го навыка главного героя.

Формат выходных данных

В единственной строке выходных данных должно содержаться целое неотрицательное число — максимальный суммарный рейтинг главного героя, который может получить Петя, используя k или менее единиц улучшения.

Задача 06. Кодзирос и Ferrari (60 баллов)

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

На протяжении 10 лет автолюбитель Кодзирос собирал деньги на свою любимую машину марки Ferrari. Наконец заветная мечта Кодзирос сбылась! Теперь Кодзирос хочет доехать до своей подруги Йоханны, чтобы похвастаться перед ней своей машиной.

Кодзирос хочет доехать до подруги, двигаясь по числовой прямой. Для простоты можно считать, что Кодзирос находится в точке 0 числовой прямой, а Йоханна — в точке с абсциссой e . В некоторых точках числовой прямой находятся бензозаправки. На каждой бензозаправке заправляют только одним видом топлива: *Регуляр-92*, *Премиум-95* или *Супер-98*. Таким образом, каждая заправка характеризуется парой целых чисел t_i и x_i — номером типа бензина и своей позицией.

Одного литра бензина достаточно, чтобы проехать ровно 1 км (эта величина не зависит от типа бензина). Бензины трех видов отличаются лишь качеством, которое по данным исследований влияет на время жизни двигателя автомобиля. Бак Ferrari вмещает ровно s литров бензина (независимо от типа бензина). В момент выезда из точки f бак Кодзирос полностью заполнен бензином марки *Супер-98*. На каждой заправке Кодзирос может заправить бак любым количеством бензина, но, конечно, ни в какой момент времени количество бензина в баке не может стать больше s литров. Обратите внимание, что в баке **может** одновременно находиться бензин разных типов. Машина может двигаться как влево, так и вправо.

Для продления времени жизни двигателя Кодзирос старается в первую очередь минимизировать количество бензина

типа *Регуляр-92*. Если существует несколько стратегий проехать из 0 в e , используя минимальный объем бензина типа *Регуляр-92*, то надо проехать так, чтобы минимизировать использованный объем бензина *Премиум-95*.

Напишите программу, которая минимизирует объем потраченного бензина типа *Регуляр-92* и объем потраченного бензина типа *Премиум-95* во вторую очередь.

Формат входных данных

В первой строке входных данных находятся три целых положительных числа e, s, n ($1 \leq e, s \leq 10^9, 1 \leq n \leq 2 \cdot 10^5$) — координата точки, в которой находится Йоханна, размер бака Fuelgati и количество бензозаправок.

В следующих n строках находится по два целых числа t_i, p_i ($1 \leq t_i \leq 3, -10^9 \leq p_i \leq 10^9$), обозначающие тип i -й бензозаправки (1 — для *Регуляр-92*, 2 — для *Премиум-95* и 3 — для *Супер-98*) и положение на координатной прямой i -й заправки. Ни в какой точке координатной прямой не находится более одной бензозаправки. Допустимо, что e может совпасть с заправкой.

Формат выходных данных

Выведите два целых числа — минимальное количество бензина типа *Регуляр-92* и типа *Премиум-95*. В первую очередь необходимо минимизировать первое значение, при разных способах достичь его надо минимизировать второе значение.

Если доехать до Йоханны никак не получится выведите строку вида «-1 -1» (две минус единицы без кавычек).

Примеры

стандартный ввод	стандартный вывод
8 4 1 2 4	0 4
9 3 2 2 3 1 6	3 3
20 9 2 1 5 2 10	-1 -1

Задача 07. Зубликанцы и мумократы (60 баллов)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

В Берляндии наступила пора выборов. Фаворитами конечно являются партии зубликанцев и мумократов. Предвыборные кампании обеих партий включают в себя многочисленные митинги на n главных площадях столицы Берляндии. На каждой из n площадей конечно может митинговать только одна партия, иначе это может привести к беспорядкам. С другой стороны обе партии подали заявки на проведение огромного количества митингов так, что на всех площадях обязательно пройдут митинги. Теперь руководству столицы предстоит распределить площади между двумя партиями.

Некоторые пары площадей соединены $(n - 1)$ двусторонними дорогами таким образом, что между любой парой площадей существует единственный способ попасть с одной площади на другую. Некоторые площади находятся на окраинах столицы, поэтому соединены дорогой лишь с одной другой площадью, такие площади называются *тупиковыми*.

Мэр столицы поручил распределить все площади между партиями таким образом, чтобы в тупиковых площадях было одинаковое количество митингов первой и второй партии.

Гарантируется, что количество тупиковых площадей города чётно.

Для предотвращения возможных конфликтов между зубликанцами и мумократам было решено минимизировать количество дорог соединяющих площади с разными партиями. Вам как разработчику отдела распределения площадей предстоит определить это наименьшее количество.

Формат входных данных

В первой строке входных данных находится целое число n ($2 \leq n \leq 5000$) — количество площадей в столице Берляндии.

В следующих $n - 1$ строках находятся пары чисел x, y ($1 \leq x, y \leq n, x \neq y$) — номера площадей, соединенных дорогой. Все площади нумеруются целыми числами от 1 до n . Гарантируется, что количество тупиковых площадей города — чётно.

Формат выходных данных

Выведите единственное число — минимальное количество дорог, соединяющих площади с митингами разных партий.

Примеры

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