

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

1. Сегодня проходит второй тур олимпиады, с результатами первого тура можно ознакомиться только после проведения второго тура. Результаты за оба тура будут просуммированы для определения результатов олимпиады. Удачи на втором туре!
2. Используйте **только** ввод с клавиатуры и вывод на экран. Никакие файлы `input.txt/output.txt` или другие использовать нельзя.
3. Не используйте в Паскале модуль `crt` (и, следовательно, функцию `clrscr`).
4. Программа **должна заканчивать** работу после вывода выходных данных. Не используйте дополнительный `read/readln` или `system("pause")` в конце программы.
5. 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.
6. 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-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++ 14 (MinGW 6) — расширение `sxx`
  - Java 8 — расширение `java`, главный класс `Solution` в пакете по умолчанию
  - 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-region>, 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 ABC должно иметь вид `abcd01.pas`, а на Free Pascal — `abcd01.pp`. Решения на Java следует именовать таким же образом, при компиляции вашей программы файл будет переименован в `Solution.java`. Поэтому главный класс должен называться `Solution` и располагаться в пакете по-умолчанию.
15. Результаты будут доступны через 1-2 дня 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 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 1. Размер монитора (20 баллов)

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

Крупная компания решила запустить новую серию прямоугольных мониторов, при этом решив, что на мониторе должно быть ровно  $n$  пикселей.

Перед вами стоит задача определить размеры прямоугольного монитора — количество  $a$  строк (рядов) из пикселей и количество  $b$  столбцов из пикселей, чтобы:

- на мониторе было ровно  $n$  пикселей,
- количество строк не превышало количество столбцов, то есть  $a \leq b$ ,
- разница  $b - a$  была минимально возможной.

### Формат входных данных

В первой строке следует целое положительное число  $n$  ( $1 \leq n \leq 10^6$ ) — количество пикселей, которые должны быть на мониторе.

### Формат выходных данных

Выведите два числа — количество рядов и количество столбцов, которые должны быть на мониторе.

## Примеры

стандартный ввод	стандартный вывод
8	2 4
64	8 8
5	1 5
999999	999 1001

## Замечание

В первом примере минимальная возможная разница равна 2, тогда на мониторе должно быть 2 ряда по 4 пикселя.

Во втором примере минимальная возможная разница равна 0, тогда на мониторе должно быть 8 рядов по 8 пикселей.

В третьем примере минимальная возможная разница равна 4, тогда на мониторе должен быть 1 ряд, состоящий из 5 пикселей.

## Задача 2. Расшифровка генома мамонта (25 баллов)

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

Расшифровка генома берляндского мамонта подходит к концу!

Одна из немногих оставшихся задач — восстановление нераспознанных нуклеотидов в найденной цепочке  $s$ . Каждый нуклеотид кодируется прописной буквой латинского алфавита: 'A', 'C', 'G' или 'T'. Нераспознанный нуклеотид кодируется знаком вопроса '?'. Таким образом,  $s$  — это строка, состоящая из букв 'A', 'C', 'G', 'T' и символов '?'.

Известно, что в расшифрованном геноме берляндского мамонта количества нуклеотидов каждого из четырех видов равны между собой.

Требуется расшифровать геном и заменить каждый нераспознанный нуклеотид на один из четырёх так, чтобы количества нуклеотидов каждого из четырёх типов стали равны между собой.

### Формат входных данных

В первой строке следует целое число  $n$  ( $4 \leq n \leq 255$ ) — длина генома.

Во второй строке следует строка  $s$  длины  $n$  — описание генома, состоящее из символов 'A', 'C', 'G', 'T' и '?'.

### Формат выходных данных

Если расшифровать геном возможно, выведите его расшифровку. Если существует много вариантов расшифровки, то выведите любой из них. Если искомой замены не существует, то выведите три знака равно подряд «===» (без кавычек).

### Примеры

стандартный ввод	стандартный вывод
8 AG?C??CT	AGACGTCT
4 AGCT	AGCT
6 ????G?	===
4 AA??	===

### Замечание

В первом примере можно заменить первый знак вопроса на букву 'A', второй знак вопроса на букву 'G', в третий — на букву 'T', тогда каждого нуклеотида в геноме будет по 2.

Во втором примере геном уже корректно расшифрован и каждого нуклеотида в нём ровно по одному.

В третьем и четвертом примерах корректно расшифровать геном невозможно.

## Задача 3. Серверы (30 баллов)

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

В лаборатории есть  $n$  серверов, каждый из которых может выполнять задания. Каждый из серверов пронумерован уникальным целым числом от 1 до  $n$ .

Известно, что в течение суток поступит  $q$  заданий,  $i$ -е из которых характеризуется тремя величинами:  $t_i$  — секунда поступления задания,  $k_i$  — количество серверов, которые нужны для его выполнения, а также  $d_i$  — продолжительность выполнения этого задания в секундах. Все  $t_i$  — различны.

Для того, чтобы выполнить  $i$ -е пришедшее задание, нужно  $k_i$  серверов, которые должны быть свободны в секунду  $t_i$ . После начала выполнения задания каждый из

них будет занят в течение ближайших  $d_i$  секунд. Таким образом, они будут заняты в секунды  $t_i, t_i + 1, \dots, t_i + d_i - 1$ . Из всех свободных серверов для выполнения будут выбраны  $k_i$  серверов с наименьшими номерами. Если в секунду  $t_i$  не хватает свободных серверов, то задание будет проигнорировано.

Напишите программу, определяющую, какие задания будут выполнены серверами, а какие будут проигнорированы.

### Формат входных данных

В первой строке следует два целых положительных числа  $n$  и  $q$  ( $1 \leq n \leq 100$ ,  $1 \leq q \leq 10^5$ ) — количество серверов и количество заданий.

В следующих  $q$  строках записаны три целых числа  $t_i, k_i$  и  $d_i$  ( $1 \leq t_i \leq 10^6$ ,  $1 \leq k_i \leq n$ ,  $1 \leq d_i \leq 1000$ ) — секунда поступления  $i$ -го задания, количество серверов, необходимых для его выполнения, а также время выполнения этого задания. Задания заданы в хронологическом порядке, и все они поступят в разные секунды.

### Формат выходных данных

Выведите  $q$  строк. Если  $i$ -е задание будет принято на выполнение, выведите в  $i$ -й строке сумму номеров серверов, на которых будет выполняться это задание. В противном случае выведите -1.

### Примеры

стандартный ввод	стандартный вывод
4 3 1 3 2 2 2 1 3 4 3	6 -1 10
3 2 3 2 3 5 1 2	3 3
8 6 1 3 20 4 2 1 6 5 5 10 1 1 15 3 6 21 8 8	6 9 30 -1 15 36

### Замечание

В первом примере в секунду 1 придет первое задание, которое будет выполняться на серверах с номерами 1, 2 и 3 (сумма номеров равна 6) в течение двух секунд.

В секунду 2 поступит второе задание, которое будет проигнорировано, так как в эту секунду будет свободен только сервер номер 4. В секунду 3 придет третье задание. К этому времени как раз освободятся серверы с номерами 1, 2 и 3, поэтому третье задание будет выполняться на всех имеющихся серверах с номерами 1, 2, 3 и 4 (сумма номеров равна 10).

В втором примере в секунду 3 придет первое задание, которое будет выполняться на серверах с номерами 1 и 2 (сумма равна 3) в течение трех секунд. В секунду 5 поступит второе задание, которое будет выполняться на сервере номер 3, так как первые два сервера будут заняты выполнением первого задания.

## Задача 4. Кошка и мышки (35 баллов)

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

На координатной прямой в точке 0 находится кошка, а правее неё находятся  $n$  мышек. Кошка преодолевает единицу расстояния за  $T$  секунд, а про каждую мышку известно время  $t_i$ , за которое  $i$ -я мышка преодолевает единицу расстояния.

В нулевой момент времени все мышки начинают убежать вправо от кошки, а кошка начинает двигаться за ними. И кошка и мышки бегут равномерно прямолинейно. Когда кошка догоняет какую-то мышку, она её сразу съедает, при этом она начинает преодолевать единицу расстояния медленнее ровно на секунду. Допустимо, что кошка может догнать двух или более мышек одновременно. В таком случае она тут же съест их всех. Если кошка в какой-то момент времени догонит  $a$  мышек, то, после того, как она их съест, время, за которое кошка преодолевает единицу расстояния, увеличится на  $a$  секунд.

Перед вами стоит задача определить, сколько мышек успеет съесть кошка, а также позиции, в которых она это сделает. Считайте, что кошка ест мышек мгновенно.

### Формат входных данных

В первой строке следует два целых положительных числа  $n$  и  $T$  ( $1 \leq n \leq 2000$ ,  $1 \leq T \leq 5000$ ) — количество мышек и время, за которое кошка преодолевает одну единицу пути.

В следующих  $n$  строках следует по два целых числа  $x_i$  и  $t_i$  ( $1 \leq x_i \leq 10^5$ ,  $1 \leq t_i \leq 5000$ ) — начальная позиция  $i$ -й мышки и время, за которое она преодолевает единицу пути.

Помните, что сначала кошка находится в точке 0.

### Формат выходных данных

В первую строку выведите количество мышек, которых успеет съесть кошка. Во вторую строку выведите в порядке неубывания координаты, в которых кошка будет есть очередную мышку (если кошка съела в какой-то точке, например, трёх мышек, то эту координату нужно выводить три раза). Выводите координаты не менее, чем с 7 знаками после десятичной точки.

### Примеры

стандартный ввод	
3 1	
3 5	
3 2	
1 2	
стандартный вывод	
2	
2.0000000000	4.3333333333

стандартный ввод	
5 5	
4 7	
4 7	
4 7	
12 9	
7 8	
стандартный вывод	
4	
14.0000000000	14.0000000000 14.0000000000 66.0000000000

### Замечание

В первом примере кошка сначала съест третью мышку в точке 2, после чего время, за которое кошка преодолевает единицу пути увеличится на 1. Затем, кошка догонит первую мышку в точке 4.3333333333, после чего время, за которое кошка преодолевает единицу пути, станет равным 3. Поэтому вторую мышку кошка догнать не сможет.

Во втором примере кошка сначала съест первых трёх мышек в точке 14, после чего время, за которое кошка преодолевает единицу пути, станет равным 8. Затем кошка съест четвертую мышку в точке 66, и время, за которое преодолевает кошка единицу пути, увеличится еще на 1. Поэтому пятую мышку кошка догнать не сможет.

## Задача 5. Зима близко (40 баллов)

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

В Берляндии зима длится  $n$  дней. Про каждый из дней известен прогноз на среднюю температуру воздуха.

У Васи есть новый комплект зимней резины, на котором можно безопасно ездить не более  $k$  дней при любой средней температуре воздуха. Через  $k$  дней использования (независимо от температуры этих дней) комплект зимней резины изнашивается. Эти  $k$  дней могут идти не подряд.

Перед первым днем зимы машина Васи всё еще на летней резине. На ней можно ездить безопасно любое количество дней, для которых средняя температура воздуха неотрицательна. Ездить на летней резине в дни, для которых средняя температура воздуха отрицательна, нельзя.

Вася может переобуwać свою машину с летней резины на зимнюю и наоборот в начале любого дня.

Найдите минимальное количество раз, которое Васе нужно переобуть свою машину с летней резины на зимнюю и с зимней резины на летнюю, чтобы безопасно проехать всю зиму. По окончании зимы машина может быть на любой резине.

### Формат входных данных

В первой строке следуют два целых положительных числа  $n$  и  $k$  ( $1 \leq n \leq 2 \cdot 10^5$ ,  $0 \leq k \leq n$ ) — количество зимних дней и количество дней, которое можно ездить на зимней резине.

Во второй строке следует последовательность из  $n$  целых чисел  $t_1, t_2, \dots, t_n$  ( $-20 \leq t_i \leq 20$ ) — средняя температура воздуха в  $i$ -й день зимы.

### Формат выходных данных

Выведите минимальное количество раз, которое нужно переобуть машину с зимней резины на летнюю и с летней резины на зимнюю, чтобы безопасно проехать всю зиму. Если это невозможно, выведите  $-1$ .

### Примеры

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

### Замечание

В первом примере Вася должен перед первым днем зимы поменять резину с летней на зимнюю, проехать на ней три дня, после чего переодеть резину с зимней на летнюю, так как безопасно на зимней резине можно ездить как раз три дня. Таким образом, суммарное количество смен резины равно двум.

Во втором примере Вася должен перед первым днем зимы поменять резину с летней на зимнюю, затем после первого дня зимы поменять резину с зимней на летнюю. После второго дня нужно вновь поменять резину с летней на зимнюю, и после третьего дня поменять резину с зимней на летнюю. Таким образом, суммарное количество смен резины равно четырём.

## Задача 6. Комментарии (45 баллов)

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

Редкая статья в интернете обходится без возможности комментирования. Вот и на сайте, который написал Поликарп, у каждой статьи есть лента комментариев.

Каждый комментарий на сайте Поликарпа — это непустая строка из строчных или прописных латинских букв. Комментарии имеют древовидную структуру, то есть у каждого комментария, кроме корневых (то есть комментариев самого верхнего уровня) есть ровно один родительский комментарий.

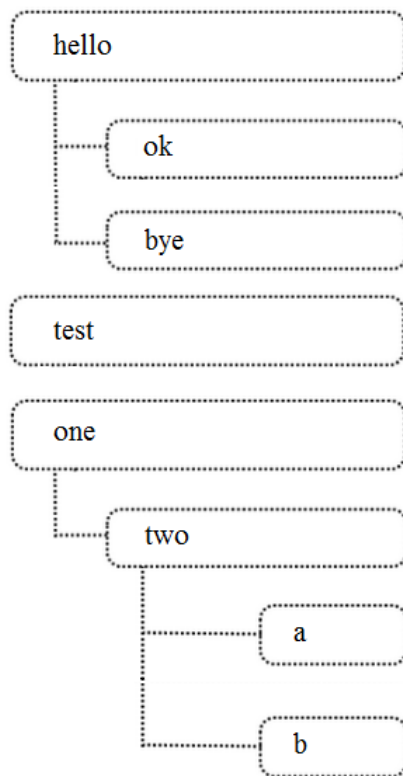
При сохранении комментариев на жесткий диск Поликарп использует следующий формат. Каждый комментарий он записывает так:

- сначала следует текст комментария;
- затем следует количество комментариев, для которых он является родительским (то есть тех, которые даны в ответ на этот комментарий);

- после этого следуют комментарии, для которых этот является родительским (запись комментариев происходит по такому же алгоритму).

Все элементы в записи выше разделяются единичными запятыми. Аналогично, комментарии первого уровня Поликарп записывает через запятую.

Например, если комментарии имели вид:



то первый будет записан как «hello,2,ok,0,bye,0», второй как «test,0», а третий как «one,1,two,2,a,0,b,0». Вся ветка комментариев будет записана как: «hello,2,ok,0,bye,0,test,0,one,1,two,2,a,0,b,0».

По заданной ветке комментариев в формате описанном выше выведите комментарии в другом формате:

- сначала выведите  $d$  — максимальную глубину вложенности комментариев;
- затем выведите  $d$  строк,  $i$ -я из них соответствует комментариям вложенности  $i$ ;
- для  $i$ -й строки выведите комментарии вложенности  $i$  в порядке их появления в записи Поликарпа, разделённые единичными пробелами.

### Формат входных данных

В первой строке следует непустая ветка всех комментариев, состоящая из строчных и прописных букв латинского алфавита, цифр и запятых. Длина строки не превышает  $10^6$  символов. Гарантируется, что заданная структура комментариев корректна.

### Формат выходных данных

Выведите комментарии в формате, описанном в условии. Для каждого уровня вложенности комментарии выводите в том же порядке, в котором они заданы во входных данных.

### Примеры

стандартный ввод	
hello,2,ok,0,bye,0,test,0,one,1,two,2,a,0,b,0	
стандартный вывод	
3	
hello test one	
ok bye two	
a b	
стандартный ввод	
a,5,A,0,a,0,A,0,a,0,A,0	
стандартный вывод	
2	
a	
A a A a A	

стандартный ввод
A,3,B,2,C,0,D,1,E,0,F,1,G,0,H,1,I,1,J,0,K,1,L,0,M,2,N,0,O,1,P,0
стандартный вывод
4 A K M B F H L N O C D G I P E J

### Замечание

Первый пример разобран в условии.

## Задача 7. Игорь и интересные числа (50 баллов)

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

Игорь очень любит 16-ричную систему счисления и считает целое **положительное** число в 16-ричной системе счисления *интересным*, если каждая цифра и буква в нём встречается не более  $t$  раз. Например, если  $t = 3$ , то числа 13a13322, aaa, abcdef0123456789 — интересные, а числа aaaa, abababab и 1000000 — нет.

Перед вами стоит задача найти  $k$ -е по счёту *интересное* для Игоря число в 16-ричной системе счисления. Число не должно содержать лидирующих нулей.

### Формат входных данных

В первой строке следует два целых числа  $k$  и  $t$  ( $1 \leq k \leq 2 \cdot 10^9$ ,  $1 \leq t \leq 10$ ) — номер искомого числа и максимальное число раз, которое может встречаться цифра или буква в этом числе.

Гарантируется, что не менее чем в половине тестов  $t = 1$ .

### Формат выходных данных

Выведите единственное целое число в 16-ричной системе счисления, которое является  $k$ -м по счёту *интересным* для Игоря числом в 16-ричной системе счисления.

### Примеры

стандартный ввод	стандартный вывод
17 1	12
1000000 2	fca2c
200000000 5	bebc69a
999999 3	f467f

### Замечание

Первые 20 чисел, которые являются *интересными*, если  $t = 1$ : 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, 10, 12, 13, 14, 15. Поэтому ответ на первый пример равен 12.

## Задача 8. Дороги (55 баллов)

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

В Берляндии есть  $n$  городов и дороги, соединяющие некоторые пары городов. Известно, что дорог в Берляндии  $n - 1$  и по ним можно добраться от любого города до любого другого.

Под простым путём между городами  $u$  и  $v$  понимается путь, в котором каждый из городов встречается не более одного раза, а начальная и конечная вершины пути равны  $u$  и  $v$ .

Президентом Берляндии было принято решение о том, что нужно построить минимальное количество дорог таким образом, чтобы между любой парой городов было как минимум два различных простых пути. Пути считаются различными, если они не имеют общих дорог, но могут иметь общие города. Запрещается строить дорогу между городами, если такая дорога между ними уже была построена.

Ваша задача — предложить искомым план постройки дорог.

### Формат входных данных

В первой строке следует целое положительное число  $n$  ( $3 \leq n \leq 2000$ ) — количество городов в Берляндии.

В следующих  $n - 1$  строках следует по два различных целых числа от 1 до  $n$  — номера городов, которые соединены очередной дорогой.

### Формат выходных данных

В первую строку выведите число  $k$  — минимальное количество дорог, которые



нужно построить, чтобы между любой парой городов было как минимум два различных простых пути.

В следующих  $k$  строках выведите по два числа — описание дорог. Каждая новая дорога должна быть выведена ровно один раз. Дороги и города в описании дорог можно выводить в любом порядке. Если решений несколько, выведите любое из них.

### Примеры

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

### Замечание

В первом примере достаточно построить две дороги, например, между городами 1 и 4 и между городами 1 и 3.

Во втором примере достаточно построить три дороги, например, между городами 1 и 5, между городами 7 и 3 и между городами 6 и 2.