

1. ПРОЦЕДУРА WRITELN. КОМПИЛЯЦИЯ ПРОГРАММ. СИМВОЛЫ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ (СРЕДА РАЗРАБОТКИ CODE::BLOCKS. СОЗДАНИЕ ПРОЕКТА. СТРУКТУРА ПРОГРАММЫ. ОПЕРАТОР ВЫВОДА. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ)	5
2. ЧИСЛА С ПЛАВАЮЩЕЙ ТОЧКОЙ (ОПЕРАЦИЯ ВЕЩЕСТВЕННОГО ДЕЛЕНИЯ)	8
3. ВЫВОД ТЕКСТОВОЙ ИНФОРМАЦИИ. ПРОЦЕДУРА READLN. ВЫЗОВ WRITELN БЕЗ ПАРАМЕТРОВ. ПРОЦЕДУРА WRITE (ОПЕРАЦИИ С БУФЕРОМ ОБМЕНА. МАНИПУЛЯТОР ENDL. КОДИРОВКИ 1251 И 866)	11
4. ПЕРЕМЕННЫЕ. ОПЕРАТОР ПРИСВАИВАНИЯ. ВЫЗОВ WRITELN С НЕСКОЛЬКИМИ ПАРАМЕТРАМИ (ПЕРЕМЕННЫЕ. ТИП LONG LONG. ОПЕРАТОР ПРИСВАИВАНИЯ)	16
5. ВВОД ЗНАЧЕНИЙ ПЕРЕМЕННЫХ С КЛАВИАТУРЫ (ОПЕРАТОР ВВОДА)	20
6. ЦИКЛЫ. ОПЕРАТОР FOR (ЦИКЛЫ. ОПЕРАТОР FOR. ОПЕРАТОР ИНКРЕМЕНТА)	24
7. ПРИМЕНЕНИЕ ПЕРЕМЕННОЙ ЦИКЛА (ПРИМЕНЕНИЕ ПЕРЕМЕННОЙ ЦИКЛА. ОПЕРАТОР СОСТАВНОГО ПРИСВАИВАНИЯ. ОПЕРАТОР ДЕКРЕМЕНТА)	28
8. АЛГОРИТМ НАКОПЛЕНИЯ СУММЫ. ОПЕРАТОРНЫЕ СКОБКИ. ПРОЦЕДУРЫ INC И DEC. ЦЕЛОЧИСЛЕННЫЕ ТИПЫ ДАННЫХ (АЛГОРИТМ НАКОПЛЕНИЯ СУММЫ)	35
9. ВЛОЖЕННЫЕ ЦИКЛЫ	39
10. ОПЕРАЦИИ DIV И MOD (ОПЕРАЦИИ ЦЕЛОГО ДЕЛЕНИЯ И ОСТАТКА ОТ ДЕЛЕНИЯ)	44
11. ВЫВОД ТАБЛИЧНОЙ ИНФОРМАЦИИ. ФОРМАТИРОВАННЫЙ ВЫВОД НА ЭКРАН (ПЕРЕМЕННАЯ-СЧЁТЧИК. ФОРМАТИРОВАННЫЙ ВЫВОД. МАНИПУЛЯТОРЫ SETW, SETFILL)	52
12. ФУНКЦИЯ SQR. ФУНКЦИЯ SQRT. ТИП EXTENDED. ФОРМАТИРОВАННЫЙ ВЫВОД ВЕЩЕСТВЕННЫХ ЧИСЕЛ. ФУНКЦИЯ ABS (ПРИВЕДЕНИЕ ТИПОВ ДАННЫХ. ТИП DOUBLE. МАНИПУЛЯТОРЫ FIXED, SETPRECISION. ФУНКЦИИ SQRT, ABS. МАТЕМАТИЧЕСКОЕ ОЖИДАНИЕ СЛУЧАЙНОЙ ВЕЛИЧИНЫ)	57
13. ТИП BOOLEAN. ОПЕРАЦИИ СРАВНЕНИЯ. ЛОГИЧЕСКИЕ ОПЕРАЦИИ AND, OR, XOR, NOT (ТИП BOOL. ПРИВЕДЕНИЕ К ТИПУ BOOL. ОПЕРАЦИИ СРАВНЕНИЯ. ЛОГИЧЕСКИЕ ОПЕРАЦИИ AND, OR, XOR, NOT. ТАБЛИЦЫ ИСТИННОСТИ. ПРИОРИТЕТ ОПЕРАЦИЙ)	62
14. ОПЕРАТОР ЦИКЛА WHILE. ОТЛАДКА ПРОГРАММЫ (ОПЕРАТОР ЦИКЛА WHILE. ОТЛАДОЧНЫЙ ВЫВОД)	68
15. ОПЕРАТОР ЦИКЛА REPEAT. АЛГОРИТМ ВЫДЕЛЕНИЯ ЦИФР ЧИСЛА (ОПЕРАТОР ЦИКЛА DO WHILE. АЛГОРИТМ ВЫДЕЛЕНИЯ ЦИФР ЧИСЛА)	70
16. ОПЕРАТОР IF. ФУНКЦИЯ RANDOM. ПРОЦЕДУРА RANDOMIZE. ОПЕРАТОР BREAK. ФУНКЦИЯ ODD (ОПЕРАТОР УСЛОВИЯ IF. ТЕРНАРНЫЙ ОПЕРАТОР. ОПРЕДЕЛЕНИЕ НЕЧЁТНОСТИ ЧИСЛА. ФУНКЦИИ RAND, SRAND, TIME. ОПЕРАТОР BREAK. ВЫХОД ИЗ ГЛАВНОГО ТЕЛА ПРОГРАММЫ)	72
17. ТИП TEXT. ЧТЕНИЕ И ЗАПИСЬ В ФАЙЛ. ПЕРЕМЕННЫЕ INPUT И OUTPUT. ПРОЦЕДУРА HALT (ФАЙЛОВЫЙ ВВОД/ВЫВОД. ФУНКЦИЯ FREOPEN)	78
18. ЛИНЕЙНЫЕ МАССИВЫ. АЛГОРИТМ ПОИСКА ЭЛЕМЕНТА. МЕТОД ФЛАГОВ. ФЛАГ ПЕЧАТИ (ЛИНЕЙНЫЕ МАССИВЫ. АЛГОРИТМ ПОИСКА ЭЛЕМЕНТА. ФЛАГ ПЕЧАТИ РАЗДЕЛИТЕЛЯ)	83
19. СОРТИРОВКА ПОДСЧЁТОМ (СОРТИРОВКА ПОДСЧЁТОМ. МЕТОДЫ УСКОРЕНИЯ ОПЕРАЦИЙ ВВОДА / ВЫВОДА)	87
20. ПОИСК НАИБОЛЬШЕГО ЭЛЕМЕНТА. КОНСТАНТЫ MAXINT, MAXLONGINT (ПОИСК НАИБОЛЬШЕГО ЭЛЕМЕНТА. ФУНКЦИИ MAX, MIN)	91

21. ОБМЕН ЗНАЧЕНИЙ ДВУХ ПЕРЕМЕННЫХ. ЦИКЛИЧЕСКИЙ СДВИГ ЭЛЕМЕНТОВ МАССИВА	98
22. СОРТИРОВКА ВЫБОРОМ. СУММА ЧЛЕНОВ АРИФМЕТИЧЕСКОЙ ПРОГРЕССИИ	100
23. ПОИСК ПОСЛЕДОВАТЕЛЬНОСТЕЙ В МАССИВЕ. МЕТОД СКАНИРУЮЩЕЙ ПРЯМОЙ. СЛОЖНОСТЬ АЛГОРИТМА	105
24. ТИП CHAR. КОДЫ СИМВОЛОВ. СИМВОЛ ОДИНАРНОЙ КАВЫЧКИ. ФУНКЦИИ CHR, ORD, PRED, SUCC. ТИП CHAR В INC, DEC, FOR. ФУНКЦИЯ UPCASE	109
25. ТИП STRING. ПУСТАЯ СТРОКА. СЛОЖЕНИЕ СТРОК. СИМВОЛЫ В СТРОКОВОЙ ПЕРЕМЕННОЙ. ПРОЦЕДУРА SETLENGTH. ФУНКЦИИ LENGTH И COPY. ФУНКЦИЯ EOF. ТИПЫ SHORTSTRING И ANSISTRING	112
26. ПОИСК СЛОВ В ТЕКСТЕ. ЛЕКСИКОГРАФИЧЕСКИЙ ПОРЯДОК. ПРОЦЕДУРА CONTINUE	117
27. ПРОЦЕДУРЫ INSERT, DELETE	120
28. ФУНКЦИЯ POS. ОГРАНИЧЕНИЕ ДЛИНЫ В СТРОКОВОЙ ПЕРЕМЕННОЙ. ТИПИЗИРОВАННЫЕ КОНСТАНТЫ. ПРОЦЕДУРЫ STR, VAL. ОПЕРАТОР IN	121
29. ДВУМЕРНЫЕ МАССИВЫ. ЗАДАЧИ С ОТКРЫТЫМИ ТЕСТАМИ	124
30. СИСТЕМА КООРДИНАТ НА ПЛОСКОСТИ. ДЛИНА ОТРЕЗКА. КООРДИНАТЫ СЕРЕДИНЫ ОТРЕЗКА. ТОЧНОСТЬ ВЫЧИСЛЕНИЙ. ПРЯМОУГОЛЬНИК. ТОЧКА ВНУТРИ ПРЯМОУГОЛЬНИКА СО СТОРОНАМИ, ПАРАЛЛЕЛЬНЫМИ ОСЯМ КООРДИНАТ. ФУНКЦИИ TRUNC, INT, FRAC. ОСТАТОК ОТ ДЕЛЕНИЯ ВЕЩЕСТВЕННЫХ ЧИСЕЛ. ВЫЧИСЛЕНИЕ СТЕПЕНИ С ВЕЩЕСТВЕННЫМ ПОКАЗАТЕЛЕМ	130
31. НЕРАВЕНСТВА ТРЕУГОЛЬНИКА. ВИДЫ ТРЕУГОЛЬНИКОВ. ПЛОЩАДЬ ТРЕУГОЛЬНИКА. МЕТОД ТРАПЕЦИЙ	133
32. КРУГ И ОКРУЖНОСТЬ. ДЛИНА ОКРУЖНОСТИ. ПЛОЩАДЬ КРУГА	134
33. ОБЪЯВЛЕНИЕ ПРОЦЕДУР И ФУНКЦИЙ. ПРОЦЕДУРА EXIT	137
34. КРАТЧАЙШЕЕ РАССТОЯНИЕ В ТАБЛИЦЕ. МАНХЭТТЕНСКОЕ РАССТОЯНИЕ	143
35. СТЕК И ДЕК	143
36. ОЧЕРЕДЬ. КОНТЕЙНЕР PAIR	145
37. КОЛЬЦЕВАЯ ОЧЕРЕДЬ. ВОССТАНОВЛЕНИЕ ПУТИ. КОНТЕЙНЕР VECTOR	151
38. РЕКУРСИЯ. ДИРЕКТИВА FORWARD	154
39. БЫСТРАЯ СОРТИРОВКА. ОБЪЯВЛЕНИЕ ТИПОВ ДАННЫХ. ИНТЕРАКТИВНЫЕ ЗАДАЧИ. ПОДКЛЮЧЕНИЕ БИБЛИОТЕК	159
40. ЖАДНЫЕ АЛГОРИТМЫ	161
41. ДИХОТОМИЯ. ТРИХОТОМИЯ. ПРЕФИКСНЫЕ СУММЫ. ПРЕФИКСНЫЕ СУММЫ НА МАТРИЦЕ	163
42. ДВОИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ. АЛГОРИТМ ГЕНЕРАЦИИ ДВОИЧНЫХ ЧИСЕЛ (ДВОИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ. АЛГОРИТМ ГЕНЕРАЦИИ ДВОИЧНЫХ ЧИСЕЛ)	165
43. ОПЕРАЦИИ СДВИГА. БЫСТРОЕ УМНОЖЕНИЕ И ДЕЛЕНИЕ НА СТЕПЕНЬ 2. ВЫДЕЛЕНИЕ ЧАСТИ ЧИСЛА. БИТОВАЯ МАСКА. БЫСТРЫЙ ОСТАТОК ОТ ДЕЛЕНИЯ НА СТЕПЕНЬ 2	167

44. ПРИЗНАКИ СТЕПЕНИ 2. ПЕРЕБОР ПОДМАСОК ЗАДАННОЙ МАСКИ. БЫСТРОЕ ВОЗВЕДЕНИЕ В СТЕПЕНЬ. ВЫЧИСЛЕНИЯ ПО МОДУЛЮ. КОД ГРЕЯ	167
45. ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ. ПЕРЕВОД ЧИСЕЛ МЕЖДУ ШЕСТНАДЦАТЕРИЧНОЙ И ВОСЬМЕРИЧНОЙ СИСТЕМАМИ СЧИСЛЕНИЯ. N-РИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ	168
46. ПЕРЕСТАНОВКИ И СОЧЕТАНИЯ	169
47. ПРИЗНАКИ ДЕЛИМОСТИ. НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ. АЛГОРИТМ ЕВКЛИДА. НАИМЕНЬШЕЕ ОБЩЕЕ КРАТНОЕ. ЧИСЛОВЫЕ ЦИКЛЫ. ПЕРИОД БЕСКОНЕЧНОЙ ДРОБИ	170
48. ПРОСТЫЕ И СОСТАВНЫЕ ЧИСЛА. АЛГОРИТМЫ ОПРЕДЕЛЕНИЯ ПРОСТОГО ЧИСЛА. РЕШЕТО ЭРАТОСФЕНА. ДВОЙНОЕ РЕШЕТО ЭРАТОСФЕНА. ФАКТОРИЗАЦИЯ ЧИСЛА. ТЕРНАРНАЯ И БИНАРНАЯ ГИПОТЕЗА ГОЛЬДБАХА	174
49. КОНТЕЙНЕРЫ SET, MAP. ИТЕРАТОРЫ	176
50. МЕТОД ДВУХ УКАЗАТЕЛЕЙ. МЕТОД ДВУХ УКАЗАТЕЛЕЙ НА МАТРИЦЕ	178
51. РЕКУРРЕНТНЫЕ СООТНОШЕНИЯ. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ. АЛГОРИТМЫ СОБИРАНИЯ СУММЫ. ЗАДАЧА О РЮКЗАКЕ. ВОССТАНОВЛЕНИЕ ПУТИ В ДИНАМИЧЕСКОМ ПРОГРАММИРОВАНИИ.....	179
52. ДВУМЕРНЫЕ РЕКУРРЕНТНЫЕ СООТНОШЕНИЯ.....	183
53. ЭЛЕМЕНТЫ ГРАФА. ВИДЫ ГРАФОВ. МАТРИЦА СМЕЖНОСТИ.....	187
54. СПИСОК СВЯЗНОСТИ. АЛГОРИТМ ФЛОЙДА. ИНДЕКСИРОВАНИЕ ВХОДНЫХ ДАННЫХ.....	190
55. АЛГОРИТМ ДЕЙКСТРЫ. ВОССТАНОВЛЕНИЕ ПУТИ В АЛГОРИТМЕ ДЕЙКСТРЫ.....	195
56. ЭЙЛЕРОВА ЦЕПЬ. ЭЙЛЕРОВ ЦИКЛ. ЭЙЛЕРОВ ГРАФ. ПОСТРОЕНИЕ ЭЙЛЕРОВА ЦИКЛА.....	200
57. СПИСОК РЕБЕР. ДВУСВЯЗНЫЕ СПИСКИ. ЭЛЕМЕНТЫ ДЕРЕВА. ГЛУБИНА УЗЛА, ВРЕМЯ ВХОДА И ВЫХОДА	201
58. МИНИМАЛЬНОЕ ОСТОВНОЕ ДЕРЕВО. СИСТЕМА НЕПЕРЕСЕКАЮЩИХСЯ МНОЖЕСТВ. АЛГОРИТМ КРУСКАЛА	202
59. НАИМЕНЬШИЙ ОБЩИЙ ПРЕДОК.....	205
60. ПРЕФИКСНОЕ ДЕРЕВО	205
61. ДЕРЕВО ОТРЕЗКОВ	206
62. N-МЕРНЫЕ РЕКУРРЕНТНЫЕ СООТНОШЕНИЯ	209
63. ПЕРСИСТЕНТНОЕ ДЕРЕВО ОТРЕЗКОВ	212
64. ПРИЗНАК РОМБА. ОБХОД ВЕРШИН ЧЕТЫРЁХУГОЛЬНИКА. ПРИЗНАК КВАДРАТА. КАНОНИЧЕСКОЕ И ОБЩЕЕ УРАВНЕНИЯ ПРЯМОЙ. ПАРАЛЛЕЛЬНОСТЬ ПРЯМЫХ. СОВПАДЕНИЕ ПРЯМЫХ.....	212
65. ВЕКТОРА. ПАРАЛЛЕЛЬНОСТЬ ВЕКТОРОВ. НОЛЬ-ВЕКТОР. НАПРАВЛЯЮЩИЙ И НОРМАЛЬНЫЙ ВЕКТОРА. УРАВНЕНИЕ ПРЯМОЙ ПО ТОЧКЕ И НАПРАВЛЯЮЩЕМУ ВЕКТОРУ. ПОСТРОЕНИЕ ПЕРПЕНДИКУЛЯРНОЙ ПРЯМОЙ	215
66. РАССТОЯНИЕ ОТ ТОЧКИ ДО ПРЯМОЙ. ПЕРЕСЕЧЕНИЕ ПРЯМЫХ. МЕТОД КРАМЕРА.....	219

67. РАСПОЛОЖЕНИЕ ТОЧЕК ОТНОСИТЕЛЬНО ПРЯМОЙ. ПРОЕКЦИЯ ТОЧКИ НА ПРЯМУЮ. ПРИНАДЛЕЖНОСТЬ ПРОЕКЦИИ ТОЧКИ ОТРЕЗКУ ПРЯМОЙ. ПОСТРОЕНИЕ СИММЕТРИЧНОЙ ТОЧКИ	220
68. ПЕРЕСЕЧЕНИЕ ОТРЕЗКА И ПРЯМОЙ. ПРИНАДЛЕЖНОСТЬ ТОЧКИ ОТРЕЗКУ. ПЕРЕСЕЧЕНИЕ ОТРЕЗКОВ. ТОЧКА ВНУТРИ ВЫПУКЛОГО МНОГОУГОЛЬНИКА. ПАРАМЕТРИЧЕСКОЕ УРАВНЕНИЕ ПРЯМОЙ. ПАРАМЕТРИЧЕСКОЕ УРАВНЕНИЕ ОКРУЖНОСТИ. КООРДИНАТЫ В ПРОСТРАНСТВЕ. ТОЧКА ВНУТРИ КЛЕТОЧНОГО МНОГОУГОЛЬНИКА	223
69. ДЛИННАЯ АРИФМЕТИКА.....	227

1. Процедура `writeln`. Компиляция программ. Символы арифметических операций (Среда разработки `Code::Blocks`. Создание проекта. Структура программы. Оператор вывода. Арифметические операции)

Первая программа на языке `Pascal` должна будет выводить на экран результат арифметического выражения $2+2$, то есть будет выступать в роли простейшего калькулятора.

Ниже приведено условие задачи.

Напишите программу, которая будет вычислять значение выражения $2+2$.



После текстового условия располагается пунктирный прямоугольник – ожидаемое состояние экрана результатов после выполнения программы.

Ниже приведено решение задачи – текст программы.

```
begin
  writeln(2+2);
end.
```

Программа состоит из трёх строк: слова "`begin`" и "`end`." начинают и заканчивают текст программы, между ними располагаются одна или несколько команд программы: операторов, процедур или функций.

В текущей программе есть только одна команда – процедура `writeln`, которая предназначена для вывода сообщений на экран. За командой указан в скобках параметр, который передаётся ей – выражение " $2+2$ ". Значение этого выражения будет вычислено и распечатано на экране.

Чтобы выполнить программу, надо преобразовать её в машинные коды – набор команд, которыми пользуется микропроцессор. Такое преобразование называется компиляцией программы

После компиляции следует передать машинные коды на исполнение микропроцессору – произвести запуск программы.

Программа после старта выводит результаты своей работы в отдельную область, которая называется экран результатов программы.

Информация на экране результатов накапливается от запуска к запуску.

Комбинации клавиш:

Ctrl-F9 – компиляция и запуск

Alt-F5 – открыть экран результатов программы

Esc – закрыть экран результатов

В арифметических выражениях на Паскале можно пользоваться операциями сложения, вычитания, умножения, деления, круглыми скобками так же, как и при вычислениях в математике.

Для операций умножения и деления на Паскале используются специальные знаки.

Ниже приведены примеры замены математических записей в команды печати результатов на Паскале.

математическая запись	команда на языке Паскаль
$3 \cdot 2$	<code>writeln(3*2);</code>
$2 (5 + 4)$	<code>writeln(2*(5+4));</code>
$8 : 4$	<code>writeln(8/4);</code>
$\frac{3+9}{2+2}$	<code>writeln((3+9)/(2+2));</code>

Знак операции (например, операции умножения) пропускать нельзя. Скобки в последнем примере с дробью используются для указания правильного приоритета операций – сначала должен быть вычислен числитель, затем – знаменатель и в конце – значение дроби.

Комбинации клавиш:

Alt-F3 – закрыть файл

F3 – открыть файл

F2 – сохранить файл

F6 – переключение между открытыми окнами

Alt-0 – показать список открытых окон

F5 – развернуть окно файла / восстановить исходный размер

Alt-X – выход из Free Pascal

Напишите программу, которая будет вычислять значение выражения $18-5+3$.

16

Вычислите значение выражения $6 \cdot 8$.

48

Вычислите значение выражения $2 \cdot (7 - 3)$.

8

Вычислите значение выражения $14 : 7$.

2.00000000E+0000

2. Числа с плавающей точкой (Операция вещественного деления)

Рассмотрим программу, которая выводит на экран значение выражения $14:7$ и на следующей строке – значение выражения $140:7$.

```
begin  
  writeln(14/7);  
  writeln(140/7);  
end.
```


2.00000000E+0000
2.00000000E+0001

В первом случае, когда вычисляется $14:7$, на экран выводится число 2, во втором, когда вычисляется $140:7$, на экран выводится число 20. Строки результатов отличаются только одной цифрой, которая стоит в конце каждой строки.


Числа, которые выведены на экран, называются числами с плавающей точкой и используются Паскалем в тех случаях, когда результат операции может быть дробным.

У числа с плавающей точкой есть две части. В первой части находятся старшие цифры числа, причём десятичная точка стоит всегда после первой цифры, а во второй части находится информация о том, на сколько позиций требуется переместить точку вправо, чтобы получить результат.


Например, в числе, представленном ниже, точку надо переместить на одну позицию вправо:

$$2.0000000000E+0001=20.0=20$$



В следующем примере **E+0002** показывает, что плавающую точку надо переместить на две позиции вправо:

$$2.0000000000E+0002=200$$



Точку надо переместить на одну позицию вправо. За точкой справа есть ненулевые цифры. Они включаются в запись числа:

$$1.2340000000E+0001=12.34$$


Ещё один пример, в котором справа от точки остаются ненулевые цифры:

$$1.2345000000E+0003=1234.5$$



Точка смещается на ноль позиций вправо. То есть, остаётся на месте:

$$2.0000000000E+0000=2$$


В случае, когда во второй части стоит отрицательное число, следует смещать десятичную точку влево. В следующем примере она перемещается на один знак влево:

$$5.0000000000E-0001=0.5$$


Ещё один пример, когда точка перемещается влево. На пустые позиции дописываются нули:

$$1.200000000E-0002 = 0.012$$


Вычислите значение выражения $1024 : 4$.

2.560000000E+0002

Вычислите значение выражения $51200 : 1.6$.

3.200000000E+0004

Вычислите значение выражения $256 : 80$.

3.20000000E+0000

Вычислите значение выражения $64 : 128$.

5.00000000E-0001

Вычислите значение выражения

$$\frac{5(14 - 8) + 6}{17 - 5}$$

3.00000000E+0000

3. Вывод текстовой информации. Процедура readln. Вызов writeln без параметров. Процедура write (Операции с буфером обмена. Манипулятор endl. Кодировки 1251 и 866)

Процедура writeln позволяет выводить на экран текстовую информацию.

Текст, который требуется распечатать, помещают в одинарные кавычки и передают его как параметр в процедуру `writeln`.

Клавиша с символом одинарной кавычки находится на клавиатуре рядом с клавишей `Enter`.

Программа, приведённая ниже, выведет на экран слово "Test".

```
begin
  writeln('Test');
end.
```



Test

Если в конце тела программы добавить вызов процедуры `readln`, то программа в момент выполнения `readln` приостановит свою работу и будет ожидать нажатия на клавишу `Enter`. После нажатия `Enter` действие программы продолжится.

Установка `readln` в конце программы позволяет не открывать экран результатов с помощью `Alt-F5` после окончания работы программы.

```
begin
  writeln('Test');
  readln;
end.
```

Во всех следующих программах помещайте последней строкой вызов процедуры `readln` для ожидания нажатия `Enter` и закрытия экрана результатов.

Процедура `writeln` на Паскале может быть вызвана без параметров. В таком случае на экране текстовый курсор перемещается на новую строку, то есть на экране образуется пустая строка.

```
begin
  writeln(1);
  writeln;
  writeln(2);
  writeln;
```

```
writeln;  
writeln(3);  
readln;  
end.
```

Программа выведет на экран числа 1, 2 и 3, разделённые пустыми строками:

```
1  
  
2  
  
3
```

Альтернативой процедуре writeln является процедура write.

Команда write также может быть использована для вывода информации на экран, однако, в отличие от writeln эта процедура не переводит курсор на новую строку экрана по окончании печати информации.

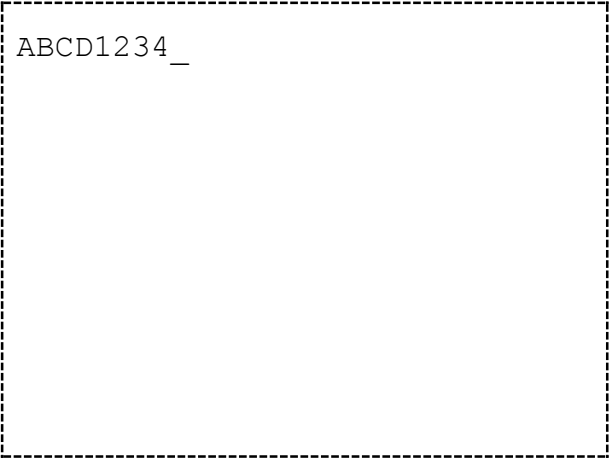
Ниже приведены фрагменты двух программ. В первом информация выводится на экран с помощью writeln, во втором - с помощью write.

```
...  
writeln('ABCD');  
writeln(1234);  
...
```

```
ABCD  
1234  
—
```

Здесь символом подчёркивания обозначено расположение курсора по окончании печати.

```
...  
write('ABCD');  
write(1234);  
...
```



ABCD1234_

Фрагмент второй программы вывел информацию обеих процедур `write` в одной экранной строке. По окончании печати курсор оставлен в конце строки.

Комбинации клавиш:

Home / End – курсор в начало / конец строки

Page Up / Page Down – курсор на страницу вверх / вниз

Ctrl-Right / Ctrl-Left – курсор на слово вперёд / назад

Ctrl-Page Up / Ctrl-Page Down – курсор в начало / конец текста

При работе с текстом программы может появиться необходимость скопировать определённый участок из одного места программы в другое. Для этих целей используют следующие операции с блоком текста:

- выделяют блок текста – подсвечивают его для дальнейшего использования;
- копируют блок текста в специальную область памяти – буфер обмена;
- перемещают текстовый курсор в требуемое место текста программы;
- копируют блок текста из буфера обмена на экран

Комбинации клавиш:

Shift-смелки – выделить блок текста

Ctrl-Insert – скопировать блок текста с экрана в буфер

Shift-Insert – вставить блок текста из буфера на экран

Shift-Delete – вырезать блок текста с экрана в буфер

Выведите на экран строку, изображённую справа.

2+2

Выведите на экран строку, изображённую справа.

--==

Выведите на экран ваши имя и фамилию на английском языке, расположив их в центре строки.

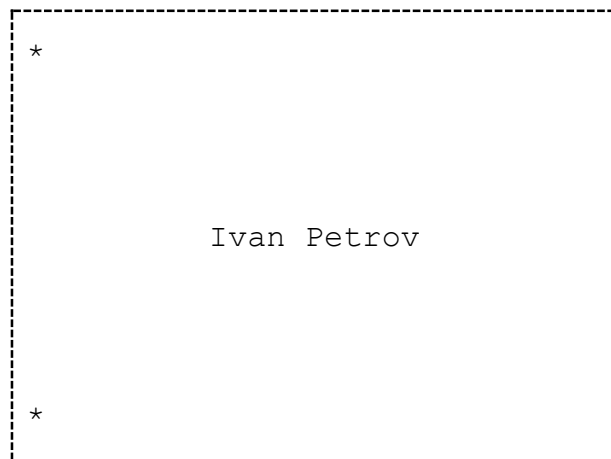
Ivan Petrov

Выведите в центре экрана ваши имя и фамилию на английском языке.

В левом верхнем углу экрана выведите символ "**".

Курсор по окончании печати расположите в левом нижнем углу экрана.

В тексте программы вывод имени и фамилии должен производиться двумя отдельными командами – одна команда выводит имя, другая – фамилию.

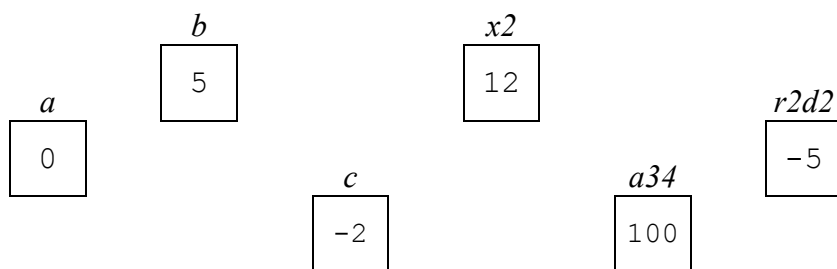


Пример фрагмента программы:

```
...
... ('Ivan');
... ('Petrov');
...
```

4. Переменные. Оператор присваивания. Вызов writeln с несколькими параметрами (Переменные. Тип long long. Оператор присваивания)

Числовая переменная – это ячейка памяти, в которой во время работы программы может храниться определённое число.



В ходе выполнения программы значение переменной может изменяться произвольное количество раз.

Присвоение переменной нового значения производится с помощью специальной команды – оператора присваивания. Например, следующий оператор присвоит переменной *a* число 5:

```
a:=5;
```

Переменные на языке Паскаль обозначаются одной или несколькими латинскими буквами и цифрами.

Для того, чтобы использовать переменные в программе, их следует объявить – сообщить Паскалю о том, как будут называться переменные, которые встретятся в программе.

Рассмотрим следующую программу:


```

var
  a,b:longint;
begin
  writeln(a);
  a:=5;
  writeln(a);
  a:=a+2;
  b:=a;
  writeln(b+3);
  readln;
end.

```

var	<i>область объявления переменных. Начинается со слова var. Располагается до главного тела программы. Здесь указываются имена и тип переменных, которые будут использоваться в программе</i>
a,b:longint;	<i>объявлены две переменные a и b типа longint (в них могут храниться целые числа из диапазона -2 147 483 648 .. 2 147 483 647)</i>
begin	<i>начало главного тела программы</i>
writeln(a);	<i>на экран выводится значение переменной a – число 0. Все переменные при старте программы заполнены нулевыми значениями</i>
a:=5;	<i>переменной a присваивается значение 5. Знак ":=" называется оператором присваивания</i>
writeln(a);	<i>на экран выводится значение переменной a – число 5</i>
a:=a+2;	<i>переменной a присваивается сумма текущего значения переменной a (числа 5) и числа 2 – число 7</i>
b:=a;	<i>переменной b присваивается значение переменной a – число 7</i>
writeln(b+3);	<i>на экран выводится сумма переменной b (числа 7) и числа 3 – число 10</i>
readln;	<i>ожидание нажатия Enter</i>
end.	<i>конец главного тела программы</i>

Ниже приведён пример вывода результатов программы.

```
0
5
10
```

Рассмотрим решение следующей задачи:

Напишите программу, в первой строке которой будет располагаться оператор присваивания переменной *a* некоторого числа.

После запуска программа должна вывести на экран поясняющую надпись "A=", а затем число, хранящееся в переменной *a*.

```
A=10
```

Пример фрагмента программы:

```
begin
  a:=10;
  ...
```

Ниже приведена программа, которая отвечает требованиям задачи:

```
var
  a:longint;
begin
  a:=10;
  write('A=');
  writeln(a);
  readln;
end.
```

```
var
  a:longint;
```

объявлена переменная a типа longint

```
begin
  a:=10;
```

согласно условию задачи в первой строке главного тела программы переменной a присваивается числовое значение

```
write('A=');  
writeln(a);  
readln;  
end.
```

на экран выводится поясняющая надпись "A=", курсор не переводится на начало новой строки

на экран выводится текущее значение переменной *a*

В программе, приведённой выше, встречаются две рядом стоящие команды `write` и `writeln`, которые выводят информацию на одну и ту же строку экрана.

Такое сочетание команд может быть заменено на одну, более эффективную команду – `writeln` с несколькими параметрами, разделёнными запятой:

```
writeln('A=', a);
```

Таким образом, программа примет вид:

```
var  
  a:longint;  
begin  
  a:=10;  
  writeln('A=', a);  
  readln;  
end.
```

В первой строке программы располагается оператор присваивания переменной *a* числового значения.

Программа должна вывести на экран надпись "A=", затем число, хранящееся в переменной *a*.

На следующей строке – надпись "A*A=", затем квадрат числа из переменной *a*.

```
A=5  
A*A=25
```

Пример фрагмента программы:

```
begin  
  a:=5;  
  ...
```

В первых двух строках программы располагаются операторы присваивания переменным a и b числовых значений.

После запуска программа должна вывести на экран три строки, содержащие информацию о значениях переменных, в формате, приведённом в примере справа.

```
Numbers: 2 and 5
Sum: 7
Product: 10
```

Пример фрагмента программы:

```
begin
  a:=2;
  b:=5;
  ...
```

В первых строках программы располагаются операторы присваивания переменным a и b числовых значений.

После запуска программа должна вывести на экран информацию о значениях переменных в формате, приведённом в примере ввода-вывода.

```
2+5=7
2*5=10
```

Пример фрагмента программы:

```
begin
  a:=2;
  b:=5;
  ...
```

5. Ввод значений переменных с клавиатуры (Оператор ввода)

Значение переменной можно вводить с клавиатуры во время работы программы, для этих целей используется процедура `readln` с передаваемым ей параметром – именем переменной, в которую вводится значение.

Рассмотрим решение следующей задачи:

Напишите программу, которая после запуска будет ожидать набора с клавиатуры произвольного целого числа, а затем будет выводить на следующей строке экрана квадрат введённого числа.

В примере ввода-вывода красным цветом выделено число, вводящееся с клавиатуры.

12
144

Ниже приведено решение задачи:

```
var
  a:longint;
begin
  readln(a);
  writeln(a*a);
  readln;
end.
```

```
var
  a:longint;
```

```
begin
  readln(a);
```

```
  writeln(a*a);
  readln;
end.
```

с клавиатуры вводится значение переменной *a*: программа ожидает набора пользователем какого-либо числа и нажатия *Enter*. После нажатия *Enter* набранное число помещается в переменную *a*

на экран выводится произведение переменной *a* на саму себя

Допускается указание в процедуре *readln* нескольких параметров через запятую – тогда во время исполнения программы с клавиатуры можно будет ввести через пробел несколько значений из одной строки.

Например, следующая команда позволит прочитать с клавиатуры значения двух переменных:

```
readln(a,b);
```

Напишите программу, которая после запуска будет ожидать набора с клавиатуры некоторого числа, а затем будет выводить на следующей строке квадрат этого числа.

В примере ввода-вывода красным цветом выделено число, вводимое с клавиатуры.

7
 $7 * 7 = 49$

Выведите на экран поясняющую надпись "A: ", затем прочитайте значение переменной *a* с клавиатуры, затем распечатайте квадрат введённого числа в формате, приведённом в примере ввода-вывода.

A: 5
 $5 * 5 = 25$

Введите значения двух переменных в формате, приведённом в примере.

Распечатайте на экране произведение этих переменных.

A=5
B=2
 $5 * 2 = 10$

Введите три числа, выведите результат согласно примеру вывода.

a, b, c: 3 2 5
 $11 = 3 * 2 + 5$

Напишите программу, которая определяет, сколько полотенец длиной a может быть получено из рулона материи длиной n ?

n, a: 13 4
3

informatics.mccme.ru

Задача №2937. Следующее и предыдущее

ej

2018-05-27 Тренировочная олимпиада для начинающих. Контекст Александра Кривошеева \

A. $2 + 2 * 2$

dl.gsu.by \ Олимпиады по информатике

GCSW \ 1999 \ Начинаящие \ День 1 \ A - "Петя и аквариум"

dl.gsu.by \ Олимпиады по информатике

Гом. обл. младшие \ 2001 \ День 2 \ 2 - "Яблоки"

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2002 \ Начинаящие, ком. \ 2 - "Рыбалка"

dl.gsu.by \ Программирование – начинающие (Р/О)
21-30 \ 5 \ 1 - "Ученики"

dl.gsu.by \ Программирование – начинающие (Р/О)
1-10 \ 6 \ 1 - "Годовой доход"

dl.gsu.by \ Программирование – начинающие (Р/О)
1-10 \ 7 \ 1 - "Тилли-Милли-Трямдия"

6. Циклы. Оператор for (Циклы. Оператор for. Оператор инкремента)

Для повторения какой-либо команды заданное количество раз в Паскале применяют операторы цикла.

Рассмотрим использование оператора цикла for на примере следующей задачи:

Используя оператор цикла for распечатайте на экране пять строк "ABCD".

```
ABCD
ABCD
ABCD
ABCD
ABCD
```

Ниже приведено решение задачи:

```
var
  i:longint;
begin
  for i:=1 to 5 do writeln('ABCD');
  readln;
end.
```

После слова for в операторе указывается имя переменной, которую называют управляющей переменной цикла. Её предназначение - изменяться от начального значения (в примере – число 1) до конечного (в примере – число 5), увеличиваясь на каждом шаге на единицу.

Ниже в таблице показаны действия, которые выполняются, когда работает оператор for:

<i>i</i>	действие	комментарий
----------	----------	-------------

1	<code>i:=1; writeln('ABCD');</code>	<i>1-й шаг цикла. Переменной i присваивается начальное значение (число 1). На экран выводится 1-я строка "ABCD"</i>
2	<code>i:=i+1; writeln('ABCD');</code>	<i>2-й шаг цикла. Переменная i увеличивается на 1. На экран выводится 2-я строка "ABCD"</i>
3	<code>i:=i+1; writeln('ABCD');</code>	<i>3-й шаг цикла. Переменная i увеличивается на 1. На экран выводится 3-я строка "ABCD"</i>
4	<code>i:=i+1; writeln('ABCD');</code>	<i>4-й шаг цикла. Переменная i увеличивается на 1. На экран выводится 4-я строка "ABCD"</i>
5	<code>i:=i+1; writeln('ABCD');</code>	<i>5-й шаг цикла. Переменная i увеличивается на 1. На экран выводится 5-я строка "ABCD"</i>

Распечатайте на экране в столбец шесть символов "#".

В программе можно использовать не более одной команды writeln.

```
#
#
#
#
#
#
```

Введите с клавиатуры значение *n*, распечатайте на экране *n* строк "ABCD".

```
n: 3
ABCD
ABCD
ABCD
```

Введите n , распечатайте строку из "-", "+", повторённых n раз.

В этой и всех последующих задачах по окончании печати переводите курсор на начало новой строки.

```
n: 4
-+-+--+
```

Введите n , распечатайте строку согласно примеру.

```
n: 4
-#=-#=-#=-#=-
```

Введите n , распечатайте строки согласно примеру.

```
n: 3
\\
  \\\
```

Введите n , распечатайте строки согласно примеру.

```
n: 3
  //\\
///    \\
```

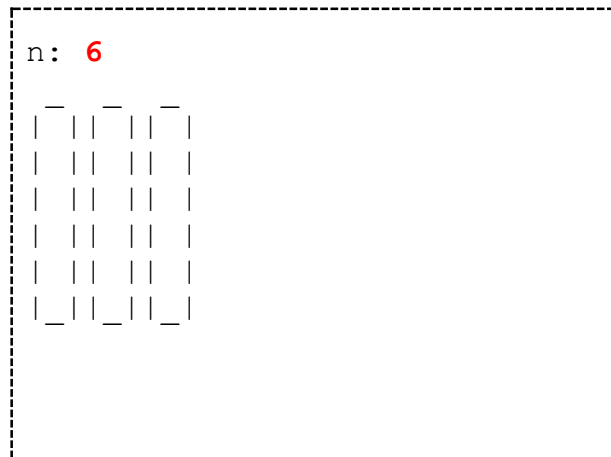
Введите n , распечатайте прямоугольник длины n согласно примеру.

```
n: 5
|_|_|_|_|
```

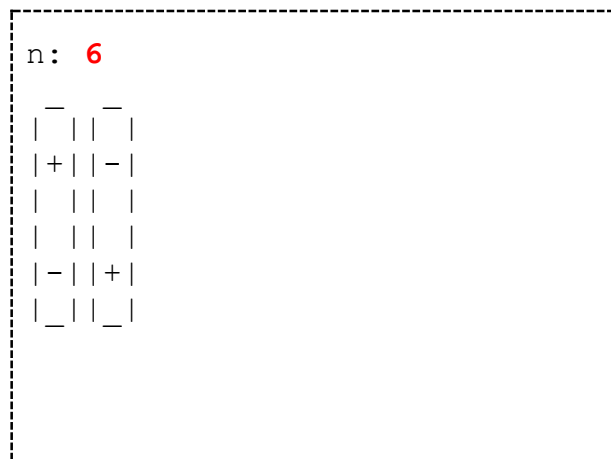
Введите n , распечатайте строки длины n согласно примеру.

```
n: 4
+ - - - +
|
|
|
|
+
```

Введите n , распечатайте три прямоугольника длины n согласно примеру.



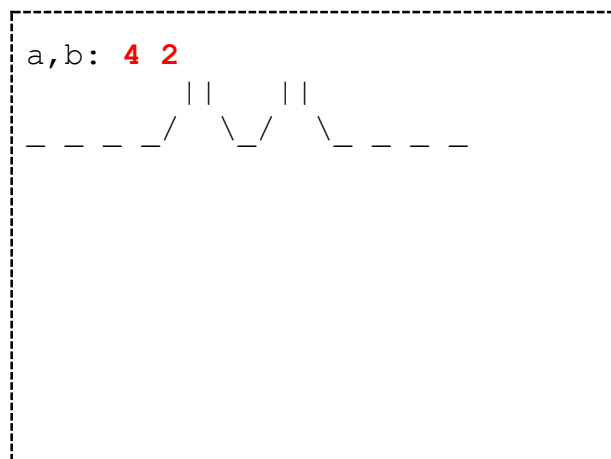
Введите n , распечатайте два изображения батареек длины n согласно примеру. $n > 3$.



Введите a, b , распечатайте строки согласно примеру.

a – длина дорожки до холмов и после них, b – количество холмов.

В программе нельзя использовать оператор *if*.



7. Применение переменной цикла (Применение переменной цикла. Оператор составного присваивания. Оператор декремента)

Рассмотрим использование переменной цикла во время работы *for* на примере следующей задачи:

Введите с клавиатуры значение n , используя оператор цикла *for* распечатайте на экране в столбец числа от 1 до n .

```
n: 6
1
2
3
4
5
6
```

Дальше приведено решение задачи и пример её исполнения для $n=6$:

```
var
  n,i:longint;
begin
  write('n: ');
  readln(n);
  for i:=1 to n do writeln(i);
  readln;
end.
```

i	действие	комментарий
1	$i:=1$; $writeln(i)$;	1-й шаг цикла. Переменной i присваивается начальное значение (число 1). На экран выводится значение i – число 1
2	$i:=i+1$; $writeln(i)$;	2-й шаг цикла. Переменная i увеличивается на 1. На экран выводится значение i – число 2
3	$i:=i+1$; $writeln(i)$;	3-й шаг цикла. Переменная i увеличивается на 1. На экран выводится значение i – число 3
...
6	$i:=i+1$; $writeln(i)$;	6-й шаг цикла. Переменная i увеличивается на 1. На экран выводится значение i – число 6

В ряде случаев для получения числовой последовательности может потребоваться использовать переменную цикла в арифметических выражениях и выводить не значение этой переменной, а значение результата арифметического выражения, в котором участвует переменная:

Используя оператор цикла *for* распечатайте на экране в столбец *n* чисел, приведённых в примере вывода.

```
n: 7
0
2
4
6
8
10
12
```

Поскольку в требуемой последовательности расстояние между числами всегда одинаково – две единицы, будем печатать на экран переменную *i*, умноженную на 2:

```
var
  n,i:longint;
begin
  write('n: ');
  readln(n);
  for i:=0 to n-1 do writeln(i*2);
  readln;
end.
```

Рассмотрим решение задачи:

Используя оператор цикла *for* распечатайте на экране в столбец *n* чисел, приведённых в примере вывода.

```
n: 6
5
7
9
11
13
15
```

Искомую последовательность чисел (5, 7, 9, 11, 13, 15) можно получить из последовательности предыдущей задачи (0, 2, 4, 6, 8, 10) путём добавления числа 5 к каждому из элементов последовательности:

```
var
  n,i:longint;
begin
  write('n: ');
```

```

readln(n);
for i:=0 to n-1 do writeln(i*2+5);
readln;
end.

```

Оператор цикла *for* позволяет изменять переменную цикла каждый шаг не только в сторону увеличения, но и в сторону уменьшения, для этого слово *to* в заголовке меняется на *downto*:

Распечатайте на экране в столбец числа от 7 до 1.

```

7
6
5
4
3
2
1

```

Решение задачи:

```

var
  i:longint;
begin
  for i:=7 downto 1 do writeln(i);
  readln;
end.

```

<i>i</i>	действие	комментарий
7	<i>i</i> :=7; writeln(<i>i</i>);	1-й шаг цикла. Переменной <i>i</i> присваивается начальное значение (число 7). На экран выводится значение <i>i</i> – число 7
6	<i>i</i> := <i>i</i> -1; writeln(<i>i</i>);	2-й шаг цикла. Переменная <i>i</i> уменьшается на 1. На экран выводится значение <i>i</i> – число 6
5	<i>i</i> := <i>i</i> -1; writeln(<i>i</i>);	3-й шаг цикла. Переменная <i>i</i> уменьшается на 1. На экран выводится значение <i>i</i> – число 5
...
1	<i>i</i> := <i>i</i> -1; writeln(<i>i</i>);	7-й шаг цикла. Переменная <i>i</i> уменьшается на 1. На экран выводится значение <i>i</i> – число 1

Распечатайте на экране в столбец числа от 1 до 5.
В программе можно использовать не более одной команды `writeln`.

```
1  
2  
3  
4  
5
```

Введите с клавиатуры n , $n \leq 9$.

Распечатайте на экране число, состоящее из n цифр 1, 2, ..., n .

```
n: 5  
12345
```

Введите с клавиатуры n , $n \leq 9$.

Распечатайте на экране число, состоящее из n цифр 1, 2, ..., n , ..., 2, 1

```
n: 5  
123454321
```


Введите с клавиатуры значение n , распечатайте на экране в строку числа от 1 до n , разделяя их пробелами.

```
n: 9  
1 2 3 4 5 6 7 8 9
```

Введите n , распечатайте строку из единиц, разделённых знаком "+", согласно примеру.

```
n: 4  
1+1+1+1=4
```

Введите a , b , распечатайте строку согласно примеру.

```
a, b: 3 5  
3*5=3+3+3+3+3
```

Введите с клавиатуры значение n , распечатайте на экране строку из n чисел, разделённых пробелами.

Числа должны изменяться согласно закону, приведённому в примере ввода-вывода.

```
n: 7
0 5 10 15 20 25 30
```

Распечатайте на экране строку из n чисел, изменяющихся по закону, приведённому в примере ввода-вывода.

```
n: 6
4 11 18 25 32 39
```

Введите n , распечатайте строки согласно примеру.

```
n: 7
1 : 7
2 : 6
3 : 5
4 : 4
5 : 3
6 : 2
7 : 1
```

Распечатайте на экране строку из n чисел, разделённых пробелами.

Числа изменяются согласно закону, приведённому в примере ввода-вывода.

```
n: 7  
0 1 4 9 16 25 36
```

8. Алгоритм накопления суммы. Операторные скобки. Процедуры *inc* и *dec*. Целочисленные типы данных (Алгоритм накопления суммы)

Рассмотрим фрагмент решения следующей задачи:

Напишите программу, которая вводит с клавиатуры значение n и распечатывает на следующей строке сумму натуральных чисел от 1 до n .

Для примера ввода-вывода сумма чисел от 1 до 5 будет вычисляться следующим образом:
 $1+2+3+4+5=15$.

```
n: 5  
15
```

```
var  
  i,n,s:longint;  
begin  
  write('n: ');  
  readln(n);  
  for i:=1 to n do s:=s+i;  
  writeln(s);  
  readln;  
end.
```

По окончании выполнения цикла в переменной s будет находиться сумма чисел от 1 до n . Рассмотрим работу цикла по шагам. Примем значение переменной n равным 5.

<i>i</i>	<i>действие</i>	<i>s</i>		<i>комментарий</i>
1	<i>s:=s+i;</i>	1	1	<i>переменной s присваивается сумма текущего значения s и числа 1</i>
2	<i>s:=s+i;</i>	3	1+2	<i>переменной s присваивается сумма текущего значения s и числа 2</i>
3	<i>s:=s+i;</i>	6	1+2+3	<i>переменной s присваивается сумма текущего значения s и числа 3</i>
4	<i>s:=s+i;</i>	10	1+2+3+4	<i>переменной s присваивается сумма текущего значения s и числа 4</i>
5	<i>s:=s+i;</i>	15	1+2+3+4+5	<i>переменной s присваивается сумма текущего значения s и числа 5</i>

Если в каждом шаге цикла требуется выполнить несколько команд, используются операторные скобки. Рассмотрим пример программы, которая читает *n* чисел и вычисляет их сумму:

```
var
  i,n,x,s:longint;
begin
  write('n: ');
  readln(n);
  for i:=1 to n do
  begin
    readln(x);
    s:=s+x;
  end;
  writeln('Sum: ',s);
  readln;
end.
```

Команды, заключённые между *begin* и *end*, выполняются оператором цикла в течение одного шага. Слова *begin* и *end* называются операторными скобками. Блок команд от *begin* до *end* называется составным оператором.

Для увеличения и уменьшения переменных более эффективно использовать специальные команды.

Ниже приведена таблица команд, которые производят быстрое увеличение и уменьшение на 1:

<i>оператор присваивания</i>	<i>команда-аналог</i>	<i>комментарий</i>
<i>x:=x+1;</i>	<i>inc (x) ;</i>	<i>переменная x увеличивается на 1</i>
<i>x:=x-1;</i>	<i>dec (x) ;</i>	<i>переменная x уменьшается на 1</i>

Увеличение и уменьшение с помощью процедур *inc* и *dec* можно производить и на числа, отличающиеся от 1, тогда у этих команд появляется второй параметр:

<i>оператор присваивания</i>	<i>команда-аналог</i>	<i>комментарий</i>
<i>x:=x+5;</i>	<i>inc (x, 5) ;</i>	<i>переменная x увеличивается на 5</i>
<i>x:=x-5;</i>	<i>dec (x, 5) ;</i>	<i>переменная x уменьшается на 5</i>

Таким образом, фрагмент программы, накапливающей сумму в переменной *s*, может быть записан в более эффективной форме:

```
...
for i:=1 to n do inc(s,i);
...
```

Комбинации клавиш:

Ctrl-F4 – встроенный калькулятор

Ниже представлены целочисленные типы данных Паскаля.

Знаковые типы (разрешено хранить отрицательные и положительные значения):

тип данных	диапазон значений	размер переменной в байтах
shortint	-128 .. 127	1
smallint	-32 768 .. 32 767	2
longint	-2 147 483 648 .. 2 147 483 647	4
int64	-9 223 372 036 854 775 808 .. 9 223 372 036 854 775 807	8

Беззнаковые типы (разрешено хранить неотрицательные значения):

тип данных	диапазон значений	размер переменной в байтах
byte	0 .. 255	1
word	0 .. 65 535	2
dword	0 .. 4 294 967 295	4
qword	0 .. 18 446 744 073 709 551 616	8

Чем больше места занимает переменная в памяти, тем медленнее она будет обрабатываться в программе.

Введите с клавиатуры неотрицательные a и b в формате, приведённом в примере ввода-вывода.

Вычислите произведение $a*b$, используя в программе только операцию сложения.

Например, произведение $2*5$ может быть представлено в виде: $2*5=2+2+2+2+2$

```
a, b: 2 5
2*5=10
```

Введите с клавиатуры значение n и выведите на экран сумму натуральных чисел от 1 до n в формате, приведённом в примере ввода-вывода.

```
n: 5
1+2+3+4+5=15
```

Введите значение n и выведите факториал числа n в формате, приведённом в примере ввода-вывода. $n \leq 20$.

Факториалом числа n называют произведение натуральных чисел от 1 до n и в математике обозначают " $n!$ ".

Например: $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$.

```
n: 5
5!=120
```

Вычислите значение выражения a^b , используя в программе только операцию умножения.

Например, 2^4 может быть представлено в виде: $2^4 = 2 \cdot 2 \cdot 2 \cdot 2$

```
a, b: 2 4
2^4=16
```

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. ком. младшие \ 2003 \ 1 - "Кефир"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2002 \ День 1 \ 2 - "Магазин"*

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинающие \ День 2 \ 4 - "Огород"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 1 - "Сумма К"

9. Вложенные циклы

В программах, где повторяющееся действие должно быть помещено в ещё один цикл, используют вложенные циклы.

Рассмотрим задачу:

Введите с клавиатуры значения *xn* и *yn*, с помощью символов "*" распечатайте прямоугольник, состоящий из *xn* столбцов и *yn* строк.

```
xn, yn: 5 3
*****
*****
*****
```

Поставим сначала перед собой более простую задачу: распечатать одну строку из *xn* символов "*".

Для печати такой строки потребуется один цикл:

```
var
  j, xn: longint;
begin
  write('xn: ');
  readln(xn);
  for j:=1 to xn do write('*');
  writeln;
  readln;
end.
```

Теперь решим более сложную задачу – распечатаем прямоугольник из *xn* столбцов и *yn* строк.

Печать одной строки надо повторить *yn* раз:

```

var
  i,j,xn,yn:longint;
begin
  write('xn,yn: ');
  readln(xn,yn);
  for i:=1 to yn do
  begin
    for j:=1 to xn do write('*');
    writeln;
  end;
  readln;
end.

```

<i>i</i>	<i>j</i>	действие	комментарий
1	1	write('*');	Стартует внешний цикл для переменной i, в его теле стартует внутренний цикл для переменной j, выполняется его первый шаг
1	2	write('*');	Цикл для j делает очередной шаг, значение i неизменно
1	3	write('*');	Цикл для j делает очередной шаг, значение i неизменно
1	4	write('*');	
1	5	write('*');	
		writeln;	Цикл для j завершил свою работу, работает очередная команда в теле внешнего цикла
2	1	write('*');	Цикл для i начинает второй шаг, цикл для j начинает свой ход с 1
2	2	write('*');	Цикл для j делает очередной шаг, значение i неизменно
...
3	5	write('*');	Цикл для j делает последний шаг
		writeln;	Цикл для i завершает последний шаг

Обратите внимание, что в конце заголовка внешнего цикла для переменной i не стоит символ ";".

Если бы заголовок оканчивался точкой с запятой, то блок команд внутри операторных скобок выполнялся бы отдельно от цикла, только один раз, уже после того, как цикл n раз выполнил бы пустой оператор ";".

действие	комментарий
for i:=1 to yn do;	Цикл для i выполняет n раз пустой оператор
begin for j:=1 to xn do write('*'); writeln; end;	По окончании цикла один раз выполняется составной оператор begin...end;

Для внешнего и внутреннего цикла следует использовать разные управляющие переменные, поскольку во время выполнения шагов внутреннего цикла работает и внешний цикл, номер шага которого должен храниться в отдельной переменной.

Комбинации клавиш:

Ctrl-Break – прерывание выполнения программы

Ctrl-F2 – выход из режима отладки

Выведите на экран согласно формату ввода-вывода квадрат, количество символов "*" в стороне которого равно n .

```
n: 3
***
***
***
```

Выведите на экран согласно формату ввода-вывода квадрат из цифр от 1 до n . $n \leq 9$.

```
n: 5
11111
22222
33333
44444
55555
```

Выведите на экран согласно формату ввода-вывода квадрат из цифр от 1 до n . $n \leq 9$.

```
n: 5
12345
12345
12345
12345
12345
```

Выведите на экран согласно формату ввода-вывода прямоугольный треугольник из цифр от 1 до n . $n \leq 9$.

```
n: 5
1
22
333
4444
55555
```

Выведите на экран согласно формату ввода-вывода прямоугольный треугольник из цифр от 1 до n . $n \leq 9$.

```
n: 5
55555
4444
333
22
1
```

Выведите на экран согласно формату ввода-вывода прямоугольный треугольник из цифр от 1 до n . $n \leq 9$.

```
n: 4
1
22
333
4444
333
22
1
```

Выведите на экран согласно формату ввода-вывода прямоугольный треугольник, количество символов "*" в сторонах которого равно n .

```
n: 5
*****
****
***
**
*
```

Выведите на экран согласно формату ввода-вывода прямоугольный треугольник, количество символов "*" в сторонах которого равно n .

```
n: 5
*****
****
***
**
*
```

Выведите на экран согласно формату ввода-вывода прямоугольный треугольник, количество символов "*" в сторонах которого равно n .

```
n: 5
   *
  **
 ***
****
*****
```

ej

2017-07-09 [Дорешивание] Тренировочная олимпиада КЮП. Контекст Вячеслава Козловского и Ивана Ревта \ F. Купить остров

10. Операции \div и \bmod (Операции целого деления и остатка от деления)

Операция \div - целого деления - предназначена для вычисления целой части результата деления двух чисел. Дробная часть результата отбрасывается. Рассмотрим процесс целого деления числа 14 на 4:

$$14 = \underbrace{1 + 1 + 1 + 1}_{4 \text{ единицы}} + \underbrace{1 + 1 + 1 + 1}_{4 \text{ единицы}} + \underbrace{1 + 1 + 1 + 1}_{4 \text{ единицы}} + \underbrace{1 + 1}_{2 \text{ единицы}}$$

Результат деления 14 на 4: 3 группы по 4 единицы

Таким образом, $14 \div 4 = 3$

Рассмотрим примеры использования \div :

x	действие	комментарий
2	$x := 5 \div 2;$	Переменной x присваивается целая часть результата деления числа 5 на 2 – число 2
3	$x := 15 \div 4;$	Число 4 содержится в числе 15 три раза. Число 3 – результат целого деления 15 на 4
3	$x := 100 \div 30;$	Ближайшее снизу от 100 число, которое делится на 30 нацело – число 90. Результат деления 90 на 30 присваивается переменной x (число 3)
5	$x := 10 \div 2;$	10 нацело делится на 2. Результат операции \div – число 5
0	$x := 7 \div 8;$	Из числа 7 число 8 можно вычесть 0 раз

Операция \div полезна при получении результата деления, который мог бы храниться в переменной целочисленного типа.

Операция остатка от целого деления mod предназначена для вычисления разницы между исходным делимым и ближайшим снизу числом, делимым нацело.

$$14 = \underbrace{1 + 1 + 1 + 1}_{\text{4}} + \underbrace{1 + 1 + 1 + 1}_{\text{4}} + \underbrace{1 + 1 + 1 + 1}_{\text{4}} + \underbrace{1 + 1}_{\text{остаток}}$$

Таким образом, $14 \text{ mod } 4 = 2$

Примеры использования mod :

<i>x</i>	<i>действие</i>	<i>комментарий</i>
1	$x := 5 \text{ mod } 2;$	Ближайшее снизу к числу 5 число, делящееся нацело на 2 - число 4. Разница между исходным 5 и делящимся нацело 4 равна 1
3	$x := 15 \text{ mod } 4;$	Если из числа 15 вычитать число 4 до тех пор, пока это возможно, то в остатке получится число 3
10	$x := 100 \text{ mod } 30;$	Если от 100 отнять 10, то получившееся число 90 будет делиться нацело на 30
0	$x := 10 \text{ mod } 2;$	10 нацело делится на 2. Результат операции mod – число 0
7	$x := 7 \text{ mod } 8;$	Из числа 7 число 8 можно вычесть 0 раз. После вычитаний останется число 7

Операция mod может быть использована для определения делимости нацело одного числа на другое:

<i>i</i>	<i>i mod 2</i>	<i>i mod 3</i>	<i>i mod 4</i>	<i>i mod 5</i>
1	1	1	1	1
2	0	2	2	2
3	1	0	3	3
4	0	1	0	4
5	1	2	1	0
6	0	0	2	1

Если $i \text{ mod } 2 = 0$, то i делится на 2 без остатка.

Если $i \text{ mod } 3 = 0$, то i делится на 3 без остатка.

Если $i \text{ mod } 4 = 0$, то i делится на 4.

Если $i \text{ mod } 5 = 0$, то i делится на 5.

В общем случае: если $i \text{ mod } x = 0$, то i делится на x .

Рассмотрим применение операций *div* и *mod* на примере решения следующей задачи:

Выведите на экран согласно примеру ввода-вывода треугольник, количество символов "*" в основании которого равно *n*.

```
n: 5
  *
 ***
*****
```

В этой программе надо организовать цикл для строк, а в каждой строке – циклы для печати пробелов и печати звёзд:

```
...
for i:=1 to количество_строк do
begin
  for j:=1 to количество_пробелов do write(' ');
  for j:=1 to количество_звёзд do write('*');
  writeln;
end;
...
```

Выясним, сколько строк должно быть в треугольнике с основанием *n*.

треугольник	есть звёзд в основании, <i>n</i>	надо строк в треугольнике	$n/2$	$n \div 2$	$(n+1) \div 2$
*	1	1	0.5	0	1
**	2	1	1	1	1
* ***	3	2	1.5	1	2
* ** ****	4	2	2	2	2
* *** *****	5	3	2.5	2	3
* ** **** *****	6	3	3	3	3

<pre> * *** ***** ***** </pre>	7	4	3.5	3	4
--	---	---	-----	---	---

Таким образом, количество строк в треугольнике вычисляется как $n/2$, округлённое до большего целого, то есть, $(n+1) \text{ div } 2$.

```

...
k:=(n+1)div 2;
for i:=1 to k do
begin
  for j:=1 to количество_пробелов do write(' ');
  for j:=1 to количество_звезд do write('*');
  writeln;
end;
...

```

Теперь возьмём треугольник с $n=7$ и найдём зависимость количества пробелов в строке от номера этой строки.

```

      *
     _ _ _
    _ _ _ _ _
   _ _ _ _ _ _
  _ _ _ _ _ _ _

```

Символом "_" обозначены пробелы.

<i>i</i> – номер строки	пробелов на строке
1	3
2	2
3	1
4	0

При $n=7$ значение $k=4$.

Если для каждой строки из k вычитать её номер i , получатся требуемые числа:

<i>i</i> – номер строки	пробелов на строке	$k-i$
1	3	3
2	2	2
3	1	1
4	0	0

Итак, количество пробелов может быть вычислено по формуле: $k-i$

```
...
k:=(n+1)div 2;
for i:=1 to k do
begin
  for j:=1 to k-i do write(' ');
  for j:=1 to количество_звёзд do write('*');
  writeln;
end;
...
```

Фрагмент программы можно оптимизировать, изменив направление внешнего цикла на убывание.

Тогда количество пробелов в строке может вычисляться по формуле $i-1$:

```
...
k:=(n+1)div 2;
for i:=k downto 1 do
begin
  for j:=1 to i-1 do write(' ');
  for j:=1 to количество_звёзд do write('*');
  writeln;
end;
...
```

Следующая оптимизация:

```
...
for i:=(n+1)div 2 downto 1 do
begin
  for j:=2 to i do write(' ');
  for j:=1 to количество_звёзд do write('*');
  writeln;
end;
...
```

Количество звёзд от строки к строке увеличивается на два, а в первой строке это количество равно 1 (для нечётных n).


```

...
z:=1;
for i:=(n+1)div 2 downto 1 do
begin
  for j:=2 to i do write(' ');
  for j:=1 to z do write('*');
  writeln;
  inc(z,2);
end;
...

```

Полученный фрагмент программы будет корректно работать для всех нечётных n .

Для чётных n в первой строке должны стоять две звезды.

n	звёзд в первой строке	$n \bmod 2$
нечётное	1	1
чётное	2	0

В таблице значения столбца звёзд увеличиваются, а значения столбца $n \bmod 2$ уменьшаются. Чтобы из $n \bmod 2$ получить количество звёзд на первой строке треугольника будем из числа 2 вычитать $n \bmod 2$:

n	звёзд в первой строке	$n \bmod 2$	$2 - n \bmod 2$
нечётное	1	1	1
чётное	2	0	2

Итоговый фрагмент:

```

...
z:=2-n mod 2;
for i:=(n+1)div 2 downto 1 do
begin
  for j:=2 to i do write(' ');
  for j:=1 to z do write('*');
  writeln;
  inc(z,2);
end;
...

```

informatics.mccme.ru

Задача №2938. Дележ яблок - 1

informatics.mccme.ru

Задача №2939. Дележ яблок - 2

Введите два числа, выведите результат согласно примеру вывода. $b < m$.

```
n, m: 13 3
13=3*a+b
a=4
b=1
```

Введите два числа, выведите результат согласно примеру вывода.

```
n, m: 14 3
14=3+3+3+3+2
```

Определите максимальное количество прямоугольных контейнеров, которое может разместиться на полу склада размером n метров на m метров.

Размеры одного контейнера: a метров на b метров. Контейнеры поворачивать нельзя.

Например, на полу размером 19 на 8 при размерах контейнера 5 на 4 может разместиться 3 ряда по 2 контейнера в каждом, всего: 6 контейнеров.

```
n, m, a, b: 19 8 5 4
6
```

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 1 - 4 кл \ 19 - "Задача 19"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 5 - 7 кл \ 8 - "Решить систему"

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2001 \ Начинающие, ком. \ 1 - "Инопланетяне"

ej

2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Андрея Мищенко и Игоря Макеев \ Н. Илья и математика

dl.gsu.by \ Программирование – начинающие (Р/О)

11-20 \ 5 \ 3 - "Школа"

Комментарий: все числа в задаче – целые, ≤ 1000

Выведите на экран согласно примеру ввода-вывода треугольник, количество символов "*" в основании которого равно n .

```
n: 5
*****
  ***
   **
    *
```

Выведите на экран согласно примеру ввода-вывода треугольник, количество символов "*" в основании которого равно n .

В программе можно использовать не более четырёх операторов цикла.

В программе нельзя использовать оператор *if*.

```
n: 7
*
**
***
****
*****
****
***
**
*
```

Выведите на экран согласно формату ввода-вывода фигуру из символов "*", количество строк и столбцов в которой всегда равно n .

В программе можно использовать не более шести операторов цикла.

В программе нельзя использовать оператор *if*.

```
n: 5
*****
***
*
***
*****
```

11. Вывод табличной информации. Форматированный вывод на экран (Переменная-счётчик. Форматированный вывод. Манипуляторы *setw*, *setfill*)

При выводе числовой информации на экран в табличной форме, как правило, в программе заводят переменную-счётчик, которая изменяется по правилу, необходимому для решения задачи (например, увеличивается каждый раз на 1, или меняется с одного числа на другое и наоборот).

В качестве примера рассмотрим решение следующей задачи:

Распечатайте на экране последовательность из n чередующихся между собой чисел 7 и 0. Первое число в последовательности: 7.

```
n: 5
7 0 7 0 7
```

```
var
  n,i,k:longint;
begin
  write('n: ');
  readln(n);
  k:=7;
  for i:=1 to n do
  begin
    write(k, ' ');
    k:=7-k;
```

```
end;
writeln;
readln;
end.
```

Здесь k играет роль переменной-счётчика, которая меняется каждый шаг с 7 на 0 и, наоборот, с 0 на 7.

При выводе табличной информации в столбцах таблицы могут оказаться разные по количеству разрядов числа, например, двухзначные и однозначные.

В независимости от количества разрядов в числе они должны выводиться с одинаковой шириной на экран, чтобы сохранить ровный столбец. Например:

```
_1 _2 _3 _4
_5 _6 _7 _8
_9 10 11 12
13 14 15 16
17 18 19 20
```

Здесь символом подчёркивания изображены дополнительные пробелы, которые должны быть вставлены при выводе однозначных чисел. Вывести подобным образом числа позволяет операция форматированного вывода чисел. Пример использования:

```
...
x:=12;
writeln(x:3);
...
```

На экран будет выведена следующая строка ("_" обозначает пробел):

```
_12
```

Здесь 3 после двоеточия – это общая длина числа, дополненного пробелами, при выводе на экран. Рассмотрим различные значения x и вывод команды `writeln(x:4)` при каждом из них:

x	<code>writeln(x:4)</code>
1	___1
12	__12
123	_123
1234	1234
1234567	1234567

Обратите внимание, что в последнем случае, когда заказанная длина в команде `writeln` меньше, чем реальная длина числа, число выводится на экран полностью, без учёта заказанной длины.

Выведите на экран числа от 1 до n , отводя на каждое из них количество символов, равное значению самого числа.

Например, число 1 должно быть распечатано в одной позиции, число 2 – в двух позициях на экране, число 3 – в трёх.

Строку завершите символом ":".

В программе можно использовать не более одного оператора цикла.

```
n: 5
1 2 3 4 5:
```

Выведите на экран таблицу из xn столбцов и yn строк с последовательно возрастающими числами.

Столбцы в таблице должны располагаться вертикально. Максимальное значение элемента таблицы: 99.

```
xn, yn: 4 5
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
```

Выведите на экран согласно формату ввода-вывода таблицу из цифр от 1 до n . $n \leq 9$.

```
n: 5
1*1 1*2 1*3 1*4 1*5
2*1 2*2 2*3 2*4 2*5
3*1 3*2 3*3 3*4 3*5
4*1 4*2 4*3 4*4 4*5
5*1 5*2 5*3 5*4 5*5
```

Выведите на экран таблицу умножения для чисел от 1 до n согласно формату в примере ввода-вывода.

$n < 10$.

```
n: 6
1  2  3  4  5  6
2  4  6  8 10 12
3  6  9 12 15 18
4  8 12 16 20 24
5 10 15 20 25 30
6 12 18 24 30 36
```

Выведите на экран таблицу из xn столбцов и yn строк с числами 7 и 0, расположенными так, как показано в примере ввода-вывода.

В программе можно использовать не более двух операторов цикла.

В программе нельзя использовать оператор *if*.

```
xn, yn: 5 7
7 7 7 7 7
0 0 0 0 0
7 7 7 7 7
0 0 0 0 0
7 7 7 7 7
0 0 0 0 0
7 7 7 7 7
```

Выведите на экран таблицу из xn столбцов и yn строк с числами 1 и 2, расположенными так, как показано в примере ввода-вывода.

В программе можно использовать не более двух операторов цикла.

В программе нельзя использовать оператор *if*.

```
xn, yn: 5 6
1 1 1 1 1
2 2 2 2 2
1 1 1 1 1
2 2 2 2 2
1 1 1 1 1
2 2 2 2 2
```

Распечатайте на экране таблицу из xn столбцов и yn строк с числами 7 и 0, в порядке, указанном в примере.

В программе можно использовать не более двух операторов цикла.

В программе нельзя использовать оператор *if*.

```
xn, yn: 3 4
7 0 7
7 0 7
7 0 7
7 0 7
```

Распечатайте на экране таблицу из xn столбцов и yn строк с числами 7 и 0, расположенными в ней в шахматном порядке.

Число в левом верхнем углу таблицы: 7.

В программе можно использовать не более двух операторов цикла.

В программе нельзя использовать оператор *if*.

```
xn, yn: 3 4
7 0 7
0 7 0
7 0 7
0 7 0
```

Выведите на экран таблицу из xn столбцов и yn строк с числами, возрастающими от 1 по порядку, расположенными так, как показано в примере ввода-вывода.

Значение yn всегда чётно.

Максимальное количество элементов в таблице: 99.

В программе можно использовать не более трёх операторов цикла.

В программе нельзя использовать оператор *if*.

```
xn, yn: 4 8
1 2 3 4
. . . .
9 10 11 12
. . . .
17 18 19 20
. . . .
25 26 27 28
. . . .
```


Выведите на экран таблицу из xn столбцов и yn строк с числами, возрастающими от 1 "зигзагообразно" - так, как показано в примере ввода-вывода.

Значение yn всегда чётно.

Максимальное количество элементов в таблице: 99.

В программе можно использовать не более трёх операторов цикла.

В программе нельзя использовать оператор *if*.

```
xn, yn: 3 6
1  2  3
6  5  4
7  8  9
12 11 10
13 14 15
18 17 16
```

12. Функция *sqr*. Функция *sqrt*. Тип *extended*. Форматированный вывод вещественных чисел. Функция *abs* (Приведение типов данных. Тип *double*. Манипуляторы *fixed*, *setprecision*. Функции *sqrt*, *abs*. Математическое ожидание случайной величины)

Функция – это команда языка Паскаль, сходная по своей работе с процедурой. Функция, подобно процедуре, может принимать один или несколько входных параметров, но, в отличие от процедуры, функция возвращает вычисленное значение – результат работы функции. Пример использования функции *sqr* показан ниже. Функция *sqr* предназначена для вычисления квадрата числа:

```
x:=sqr(5);
```

В результате выполнения этой команды переменной *x* будет присвоено число 25 – квадрат числа 5, которое передавалось на вход функции *sqr*.

Результат функции может быть присвоен переменной, выведен на экран или использован в арифметическом выражении. То есть, результат функции может использоваться в тех же местах, где и обычная переменная. В следующем примере команда *writeln* выведет на экран результат арифметического выражения, в котором используется *sqr*:

```
writeln(50-sqr(4)*2);
```

Функция *sqrt* – это функция, обратная по своему действию функции *sqr*. Она вычисляет квадратный корень числа. В следующем примере на экран будет выведен результат вычисления корня из 16 – число 4.0:

```
writeln(sqrt(16));
```

Функция *sqrt* возвращает не целый, а вещественный результат, поэтому попытка присвоения результата этой функции переменной целого типа приведёт к ошибке компиляции. Следующий фрагмент программы будет вызывать ошибку во время компиляции:

```
var
```

```

a:longint;
begin
  a:=sqrt(16);
...

```

Для хранения вещественных чисел будем использовать тип данных *extended*. Переменные типа *extended* предназначены для хранения вещественных чисел. Следующий фрагмент программы будет скомпилирован успешно:

```

var
  x:extended;
begin
  x:=3.14;
  writeln(x)
  x:=sqrt(16);
...

```

Кроме достоинства переменных типа *extended* – возможности хранить вещественные значения – у них существует ряд недостатков:

- переменные типа *extended* занимают большой объём в памяти по сравнению с переменными типа *longint* – 10 байт против 4 байт;
- обработка переменных типа *extended* при сложении и вычитании медленнее, чем переменных целого типа;
- переменные типа *extended* нельзя использовать как управляющие переменные в циклах, как начальное или конечное значение цикла;
- переменные типа *extended* нельзя обрабатывать процедурами *inc* и *dec*.

Рекомендуется для ускорения программы использовать *extended* только в тех случаях, когда это действительно необходимо, а в остальных ситуациях применять целые типы данных.

Для вывода вещественных чисел в формате с фиксированной точкой используйте форматированный вывод вещественных чисел. Его использование подобно форматированному выводу целых чисел.

В форматированном выводе вещественных чисел используется два параметра – общая длина числа и количество знаков после десятичной точки. Рассмотрим несколько команд, выводящих вещественные числа в форматированном виде.

команда	результат вывода	комментарий
<code>x:=13.68;</code>		переменной <i>x</i> присваивается вещественное значение 13.68
<code>writeln(x:7:2);</code>	__13.68	число выводится в 7 позициях по длине, количество цифр после точки: 2
<code>writeln(x:7:3);</code>	_13.680	количество цифр после точки: 3, справа добавлен незначимый 0

<code>writeln(x:7:1);</code>	___13.7	количество цифр после точки: 1, дробная часть округлена
<code>writeln(x:7:0);</code>	_____14	количество цифр после точки: 0, точка не выводится
<code>writeln(x:0:1);</code>	13.7	количество цифр после точки: 1, число выводится в минимальной длине, без дополнения пробелами слева
<code>writeln(sqrt(25):7:2);</code>	___5.00	параметры форматированного вывода указываются после функции, результат которой выводится на экран

Обратите внимание, что в последнем случае, когда заказанная длина в команде `writeln` меньше, чем реальная длина числа, число выводится на экран полностью, без учёта заказанной длины.

Функция `abs` используется для вычисления абсолютной величины числа - модуля числа, то есть, положительной части числа. В следующем примере переменной `x` будет присвоено значение 5.

```
x:=abs(-5);
```

Функцией `abs` можно пользоваться в тех случаях, когда требуется получить последовательность чисел, симметричную относительно центра. В следующем примере на экран будет выведен столбец чисел 4..0..4.

```
for i:=-4 to 4 do writeln(abs(i));
```

```
4
3
2
1
0
1
2
3
4
```

Напишите программу, которая вычисляет квадратный корень вещественного числа x и выводит его согласно примеру ввода-вывода.

Точность вещественных чисел при выводе: два знака после десятичной точки.

```
x: 16
sqrt(16.00)=4.00
```

*dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 10 \ 2 - "Налоговая инспекция"*

Распечатайте на экране таблицу квадратных корней из элементов таблицы умножения – на пересечении i -го столбца и j -й строки находится квадратный корень из числа $i*j$.

Например, на пересечении 2-го столбца и 8-й строки будет находиться квадратный корень из 16 – число 4.00.

Числа в таблице выводите в следующем формате: два знака после десятичной точки, два знака – до неё. Максимальное значение n : 12.

```
n: 3
1.00  1.41  1.73
1.41  2.00  2.45
1.73  2.45  3.00
```

Распечатайте на экране последовательность чисел от n до 1 и далее до n согласно примеру ввода-вывода.

В программе используйте не более одного оператора *for*.

В программе нельзя использовать операторы *if* и *case*.

```
n: 5
5 4 3 2 1 2 3 4 5
```

Распечатайте последовательность $0..n..0$ согласно примеру ввода-вывода.

В программе используйте не более одного оператора *for*.

В программе нельзя использовать операторы *if* и *case*.

```
n: 4
0 1 2 3 4 3 2 1 0
```

Распечатайте последовательность $1..n..1$ согласно примеру ввода-вывода.

В программе используйте не более одного оператора *for*.

В программе нельзя использовать операторы *if* и *case*.

```
n: 5
1 2 3 4 5 4 3 2 1
```

Выведите на экран согласно формату ввода-вывода фигуру из символов "*", у которой длина первой и последней строки равна n .

В программе используйте не более двух операторов цикла.

В программе нельзя использовать операторы *if* и *case*.

```
n: 4
****
***
**
*
**
***
****
```

Выведите на экран согласно примеру ввода-вывода треугольник, количество символов "*" в основании которого равно n .

Значение n всегда нечётно.

В программе используйте не более двух операторов цикла.

```
n: 7
*
**
***
****
***
**
*
```

*dl.gsu.by \ Программирование – начинающие (Р/О)
21-30 \ 4 \ 1 - "Яблоки"*

*dl.gsu.by \ Программирование – начинающие (Р/О)
21-30 \ 1 \ 1 - "Детский дом"*

Комментарий: округляйте результат в меньшую сторону

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Профессионалы, ком. \ 10 - "Радикалы"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. ком. младшие \ 2001 \ 3 - "Задача о рыбе в пруду"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2002 \ Младшая группа \ 6 - "Уравнение"*

*ej
2018-05-27 Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \
В. Казино*

13. Тип *boolean*. Операции сравнения. Логические операции *and*, *or*, *xor*, *not* (Тип *bool*. Приведение к типу *bool*. Операции сравнения. Логические операции *and*, *or*, *xor*, *not*. Таблицы истинности. Приоритет операций)

Тип данных *boolean* – это логический тип данных, который предназначен для хранения двух состояний: "истина" и "ложь". В Паскале эти состояния обозначаются логическими константами – служебными словами *true* и *false*.

Пример использования переменной типа boolean:

```
Var  
  f:boolean;  
begin  
  writeln(f);  
end.
```

FALSE

Логическим переменным можно присваивать константы true или false:

```
...  
  f:=true;  
  writeln(f);  
...
```

TRUE

В Паскале существуют операции сравнения – команды, которые позволяют сравнить две части выражения, и возвращают результат сравнения – "истина" или "ложь". Их результат можно присваивать логическим переменным или использовать при печати на экране:

```
...  
  f:=2<5;  
  writeln('2<5: ',f);  
  f:=2>5;  
  writeln('2>5: ',f);  
  f:=2=1+1;  
  writeln('2=1+1: ',f);  
  writeln('2<>3: ',2<>3);  
  writeln('2>=3: ',2>=3);  
  writeln('2<=2: ',2<=2);  
...
```

2<5: TRUE
2>5: FALSE
2=1+1: TRUE
2<>3: TRUE
2>=3: FALSE
2<=2: TRUE

С переменными логического типа можно производить операции, аналогичные арифметическим операциям сложения, вычитания, умножения и деления над числами. Такие операции называют логическими.

Пусть a , b , c – переменные логического типа. Тогда будет верной следующая запись:

$c := a \text{ and } b;$

Здесь *and* – операция логического "И". На вход этой операции передаются два операнда – a и b , каждый может принимать значения *true* и *false*. В зависимости от комбинации значений входных операндов на выходе операция возвращает *true* или *false*.

Поведение логической операции можно описать в таблице, где указаны все возможные комбинации входных данных и соответствующие им результаты. Такая таблица называется таблицей истинности – она показывает, в каких случаях операция возвращает результат "истина":

a	b	c
false	false	false
false	true	false
true	false	false
true	true	true

Операция *and* – строгая операция – она возвращает *true* только в случае, когда оба аргумента истинны.

Следующая логическая операция – *or* – логическое "ИЛИ". Это нестрогая операция, она возвращает *true*, если хотя бы один операнд равен *true*:

$c := a \text{ or } b;$

a	b	c
false	false	false
false	true	true
true	false	true
true	true	true

Операция *xor* – *exclusive or* – "исключающее ИЛИ". Это операция, которая более строга, чем *or*, но менее – чем *and*. Она возвращает *true*, если ровно один операнд равен *true*:

$c := a \text{ xor } b;$

<i>a</i>	<i>b</i>	<i>c</i>
<i>false</i>	<i>false</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>

Xor возвращает *true*, если операнды не равны друг другу.

Операция *not* – логическое "НЕ". Это унарная операция – она принимает только один операнд, в отличие от предыдущих – бинарных операций:

c := not a;

<i>a</i>	<i>c</i>
<i>false</i>	<i>true</i>
<i>true</i>	<i>false</i>

Эта операция инвертирует результат – вместо *true* возвращает *false* и наоборот.

У логических операций, так же, как у арифметических сложения и умножения, есть приоритет выполнения в составном выражении:

1. *not*
2. *and*
3. *or, xor*

В выражении без скобок раньше выполняется *not*, последними – *or* и *xor*. Если требуется выполнить раньше операцию с меньшим приоритетом, используйте скобки:

c := (a or b) and c;

Напишите программу, которая вводит с клавиатуры число a , а затем выводит на экран "TRUE", если $a > 0$, иначе – "FALSE".

В программе нельзя использовать оператор *if* и операторы цикла.

```
a: -5  
FALSE
```

Напишите программу, которая вводит с клавиатуры числа a и b , затем на следующей строке запрашивает и вводит с клавиатуры результат произведения $a*b$.

В случае, если результат произведения был введён верно, программа должна распечатать на экране "TRUE", иначе – "FALSE".

В программе нельзя использовать оператор *if*.

```
a, b: 2 5  
2*5=10  
TRUE
```

Напишите программу согласно примеру, приведённому ниже.

Пользуясь программой подготовьтесь к тесту на знание логических операций: устно вычисляйте результат операции, затем запускайте программу, сравнивайте ваш результат с результатом программы.

```
FALSE
```

Пример фрагмента программы:

```
var  
  a,b:boolean;  
begin  
  a:=true;  
  b:=false;  
  writeln(a and b);  
  ...
```

Введите числа a и b , затем введите результат произведения $a*b$, затем – результат суммы $a+b$.

Выведите "TRUE", если на оба запроса результатов были даны верные ответы, иначе – "FALSE".

В программе нельзя использовать оператор *if*.

```
a,b: 2 5
2*5=10
2+5=8
FALSE
```

Введите n , a , b , результат $a*b$, результат $a+b$ согласно формату в примере ввода-вывода.

Выведите "TRUE", если количество верных результатов не меньше, чем n , иначе – "FALSE".

Возможные значения n : 1,2.

В программе нельзя использовать оператор *if*.

```
n: 1
a,b: 2 5
2*5=10
2+5=8
TRUE
```

Определите по номеру года n , является ли он високосным. Выведите "TRUE", если год високосный, иначе – "FALSE".

В современном календаре каждый год, номер которого делится на 4, является високосным, за исключением тех, которые делятся на 100 и не делятся на 400.

Например, 1900 год – не високосный, а 2000 – високосный.

В программе нельзя использовать оператор *if*.

```
n: 1900
FALSE
```

14. Оператор цикла *while*. Отладка программы (Оператор цикла *while*. Отладочный вывод)

Напишите программу, которая последовательно вводит ненулевые числа с клавиатуры в формате, приведённом в примере ввода-вывода.

При наборе числа 0 программа заканчивает ввод и выводит на экран сумму всех введённых чисел.

```
x=3
x=2
x=5
x=1
x=4
x=0
Sum: 15
```

Распечатайте на экране с помощью цикла *while* числа от 9 до 1.

```
9 8 7 6 5 4 3 2 1
```

Распечатайте на экране с помощью одного цикла *while* числа от -2.00 до 1.00 с шагом 0.5 в формате, приведённом в примере ввода-вывода.

В программе используйте не более одной процедуры *writeln*.

В программе нельзя использовать оператор *if*.

```
-2.00
-1.50
-1.00
-0.50
0.00
0.50
1.00
```

Вычислите результат целочисленного деления числа a на число b .

В программе можно использовать только операции сложения и вычитания.

```
a,b: 13 4
13 div 4=3
```

Вычислите результат целочисленного деления числа a на число b и остаток от этого деления.

В программе можно использовать только операции сложения и вычитания.

```
a,b: 13 4
13 div 4=3
13 mod 4=1
```

Произведение первых n нечётных натуральных чисел равно p .

По известному p определите n .

Например, для $p=15$ значение n равно 3:

$$15 = \underbrace{1 \cdot 3 \cdot 5}_{3 \text{ числа}}$$

В программе нельзя использовать оператор *if*.

```
p: 15
3
```

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинаящие \ День 1 \ 1 - "Помощь математику"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 5 - 8 кл \ 8 - "Бабушкины яблоки"*

codeforces.ru \ Архив

791A. Мишка и старший брат

15. Оператор цикла *repeat*. Алгоритм выделения цифр числа (Оператор цикла *do while*. Алгоритм выделения цифр числа)

Введите с клавиатуры число x и распечатайте на экране его младшую цифру. $0 \leq x \leq 2\,000\,000\,000$.

x: 123456
6

Введите с клавиатуры числа x, n и распечатайте на экране цифру, которая находится в числе x в разряде с номером n .

Разряды в числе нумеруются справа налево, начиная с нуля.

Например, в числе 123456 разряд №0 занимает цифра 6, разряд №1 – цифра 5, разряд №2 – цифра 4. $0 \leq x \leq 2\,000\,000\,000$.

x: 123456
n: 2
4

Напишите программу, которая вычисляет количество цифр в числе x . $0 \leq x \leq 2\,000\,000\,000$.

В программе можно использовать не более одного оператора цикла.

В программе нельзя использовать оператор *if*.

x: 78901
5

Введите с клавиатуры значение x , а затем распечатайте число, полученное из x путём зеркальной перестановки цифр. $0 \leq x \leq 2\,000\,000\,000$.

В программе не разрешено использование оператора *if*. В программе можно использовать не более одного оператора цикла.

x: 123456
654321

Вычислите сумму цифр числа x . $0 \leq x \leq 2\,000\,000\,000$.

x: 3456
18

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Профессионалы, ком. \ 6 - "Телепузики"

16. Оператор if. Функция random. Процедура randomize. Оператор break. Функция odd (Оператор условия if. Тернарный оператор. Определение нечётности числа. Функции rand, srand, time. Оператор break. Выход из главного тела программы)

Введите с клавиатуры значения a , b , результат $a*b$.

Выведите на экран "Right", если введённый результат равен произведению $a*b$, иначе распечатайте "Wrong".

```
a,b: 5 3
5*3=10
Wrong
```

ej

2017-08-13 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ А. Внимательность

Введите с клавиатуры значения a , b .

Выведите на экран числа a и b с соответствующим знаком операции сравнения между ними: ">", "<" или "=".

```
a,b: 5 3
5>3
```

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2001 \ Младшая группа \ 2 - "Рыбки"

Введите с клавиатуры значение n , распечатайте n случайных целых чисел из диапазона 1..5.

Значения случайных чисел, выведенные при запуске программы, не должны совпадать со значениями, полученными во время предыдущего запуска.

В программе нельзя использовать оператор if.

В программе нельзя использовать более одного оператора цикла.

```
n: 7  
1 4 3 4 2 5 1
```

Напишите программу, которая предлагает ввести результат произведения двух случайных чисел, взятых из диапазона 0..9, а затем выводит "Right", если результат введен верно, и "Wrong" – в противном случае.

```
7*9=56  
Wrong
```

Введите n , a , b , результат $a*b$, результат $a+b$, результат $a-b$ согласно формату в примере ввода-вывода.

Выведите "TRUE", если количество верных результатов не меньше, чем n , иначе – "FALSE".

Возможные значения n : 1,2,3.

```
n: 2  
a, b: 5 2  
5*2=10  
5+2=1  
5-2=3  
TRUE
```

Медведь и малина

Автор: Андрей Мищенко

Медведь Миша пришёл на свою любимую полянку, и увидел, что так много малины он не съест. Всего было n ягод. Миша решил есть ягоды, размер которых не меньше k . Ваша задача: вывести, какое количество ягод съест Медведь.

Формат ввода:

n k количество ягод и минимальный размер ягод
 $a_1 a_2 \dots a_n$ все размеры ягод

Формат вывода: количество съеденной малины.

```
5 4
1 2 5 4 11
3
```

ej

2017-09-03 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ А. Поход

ej

2017-06-11 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Игоря Макеенко и Андрея Мищенко \ А. Олимпиада в Байтландии

ej

2017-09-03 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ В. Броня для Serega

dl.gsu.by \ Программирование – начинающие (P/O)
 21-30 \ 3 \ 1 - "Четные числа"

dl.gsu.by \ Программирование – начинающие (P/O)
 1-10 \ 1 \ 1 - "Количество отличников"

dl.gsu.by \ Программирование – начинающие (P/O)
 1-10 \ 1 \ 2 - "Баскетбольная команда"

dl.gsu.by \ Программирование – начинающие (P/O)
 1-10 \ 3 \ 2 - "Модельное агентство"

dl.gsu.by \ Программирование – начинающие (P/O)
 1-10 \ 5 \ 1 - "Футбольный клуб"

dl.gsu.by \ Программирование – начинающие (Р/О)
11-20 \ 5 \ 1 - "Армия"

dl.gsu.by \ Программирование – начинающие (Р/О)
11-20 \ 2 \ 1 - "Работа"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2002 \ Младшая группа \ 1 - "Лимоны"

dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2001 \ День 1 \ 3 - "Стрельба - 1"

Напишите программу, которая n раз запрашивает результат произведения случайных чисел из диапазона 0..9, а затем выводит оценку – число из диапазона 0..10, рассчитанное пропорционально по правилу: 0 правильных ответов – оценка 0, n правильных ответов – оценка 10.

Оценка должна быть округлена до ближайшего целого числа.

```
n: 4
5*3=15
3*2=6
5*1=6
2*6=6
Score: 5
```

dl.gsu.by \ Олимпиады по информатике
GCSW \ 1999 \ Начинающие \ День 1 \ С - "Лесник"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2002 \ Младшая группа \ 2 - "Марс"

dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2001 \ День 2 \ 3 - "Змеи"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 2 \ 1 - "Мистер Бин"

Напишите программу, которая находит минимальное натуральное число, сумма цифр которого равна x . $x \leq 50$.

Например, у следующих натуральных чисел сумма цифр равна 30: 3999, 4998, 9399, ...
Минимальное из них – число 3999.

x: 30
3999

Распечатайте на экране в порядке возрастания все трёхзначные числа, сумма цифр которых равна x .

x: 3
102
111
120
201
210
300

ej

2017-08-13 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ D. Сергей и конфеты

ej

2017-06-18 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ C. Сложность обучения

codeforces.ru \ Архив

427A. Полицейские-рекруты

dl.gsu.by \ Олимпиады по информатике

GCSW \ 1999 \ Начинаящие \ День 1 \ G - "Автоматический зонд"

Напишите программу, которая определяет, является ли число x номером "счастливого" билета. $0 \leq x \leq 2\,000\,000\,000$.

Номером "счастливого" билета называется такое число, в котором сумма цифр левой половины равна сумме цифр правой половины. Если количество цифр в числе нечётно, то среднюю цифру не учитывают ни в одной из частей. Например, число 1239015 – номер "счастливого" билета. Здесь $1+2+3=0+1+5$.

```
x: 1239015
TRUE
```

Распечатайте на экране в порядке возрастания все номера "счастливых" билетов от a до b .

```
a, b: 990 1200
999
1001
1010
1102
1111
1120
```

ej

2018-05-27 Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ К. Сколько нулей?

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 10 - "Счастливые пары"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 2 - "Числа Z"

ej

2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Андрея Мищенко и Игоря Макеенко \ А. Интересная Игра

ej

2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Андрея Мищенко и Игоря Макеенко \ G. Игорь и арбузы

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 5 - "Расчёты с должниками"

codeforces.ru \ Архив

384A. Кодер

codeforces.ru \ Архив

705A. Халк

codeforces.ru \ Архив

369A. Валера и тарелки

codeforces.ru \ Архив

432A. Выбор команд

[ej](#)

2018-05-27 Тренировочная олимпиада для начинающих. Контекст Александра Кривошеева \

I. Дом, милый дом

codeforces.ru \ Архив

828A. Столы в ресторане

17. Тип *text*. Чтение и запись в файл. Переменные *input* и *output*. Процедура *halt* (Файловый ввод/вывод. Функция *freopen*)

A+B

Автор: Руслан Кулицкий

[ej \ 17-01](#)

Напишите программу, которая из файла *input.txt* читает два числа типа *longint*, а затем выводит в файл *output.txt* их сумму.

input.txt

2 3

output.txt

5

astp.ru \ Архив задач

0025 – "Больше-меньше"

[астр.ru \ Архив задач](#)
0108 – "Неглухой телефон"

Сумма чисел

Автор: Дмитрий Белойчук
[ej \ 17-02](#)

Напишите программу, которая читает из первой строки файла ввода значение n – количество чисел, затем со второй строки читает цепочку из n чисел в интервале $-10..10$ и выводит в файл вывода сумму чисел этой цепочки. $0 \leq n \leq 100000$.

input.txt

5
2 7 0 2 1

output.txt

12

[dl.gsu.by \ Олимпиады по информатике](#)
GCSW \ 2000 \ Личные соревнования \ Начинающие \ День 2 \ А - "Секретная база"

Столбец чисел

Автор: Иван Киринский
[ej \ 17-03](#)

Есть n листов бумаги, выложенных в столбец. На листах написаны различные числа. Егору необходимо переложить эти листы в ряд, но он не знает, как это сделать. Помогите Егору.

Формат ввода:

n $n \leq 100\,000$
 a_1 a – число, написанное на листе, $0 \leq a \leq 10\,000$
...
 a_n

Формат вывода: числа в строку.

input.txt

5
7
2
8
4
9

output.txt

7 2 8 4 9

[астр.ru \ Архив задач](#)
0004 – "Игра"

Самостоятельная работа

Автор: Владислав Собин

[ej \ 17-04](#)

У Пети самостоятельная работа по математике. И у него не получается решить задачу. Пете дано 3 числа. Надо определить, расположены ли эти числа в порядке неубывания – каждое число не меньше предыдущего. Помогите Пете.

Формат ввода:

$a\ b\ c$ числа в диапазоне от -2 147 483 648 до 2 147 483 647

Формат вывода: "Yes", если ответ на задачу положительный. "No", если ответ на задачу отрицательный.

input.txt

2 76 24

output.txt

No

[dl.gsu.by \ Олимпиады по информатике](#)

[GCSW \ 2000 \ Командные соревнования \ Командный тур \(начинающие\) \ А - "Плотник"](#)

Комментарий: в задаче полагайте, что $N \leq 1000$

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская обл. \ 2011 \ День 2 \ 1 - "Занимательная математика"](#)

Считалочка

Автор: Алексей Ропан

[ej \ 17-05](#)

В детском садике дети любят игру "считалочка". Все дети становятся в круг и начинают отсчитывать считалочку с определенного ребенка. На ком остановилась считалка - тот выбывает, а считалочку начинают отсчитывать со следующего по кругу. Считать заканчивают, когда остаётся один ребенок.

Юный программист Вовочка заметил, что всегда считается одна и та же считалочка. Он заранее посчитал, кто останется, если начинать с первого. Но все остальные варианты сосчитать не успел. Помогите Вовочке определить номер последнего ребенка, если счёт начнётся с k -го ребенка.

Формат ввода:

n количество детей ($1 \leq n \leq 10^9$)

m номер оставшегося ребенка при счёте с первого ($1 \leq m \leq n$)

k номер ребенка, с которого начинается счёт ($1 \leq k \leq n$)

Формат вывода: номер последнего ребенка, если счёт начнётся с k -го.

calc.in5
4
2**calc.out**

5

Цепочки одинаковых чисел

Автор: Олег Любинский

ej \ 17-06

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n чисел.

Выведите в файл по порядку все непрерывные цепочки одинаковых чисел согласно примеру вывода.

input.txt8
7 7 1 1 1 5 4 4**output.txt**7 7
1 1 1
5
4 4**Возрастающая последовательность**

Автор: Василий Шмидт

ej \ 17-07

Прочитайте из первой строки n , $n \leq 10$, со второй строки – n чисел.

Выведите в файл по порядку все непрерывные неубывающие последовательности чисел согласно примеру вывода.

input.txt8
1 7 3 6 9 9 5 1**output.txt**1 7
3 6 9 9
5
1

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2000 \ Личные соревнования \ Начинающие \ День 1 \ С - "Скалолаз"

Одинаковые числа

Автор: Дмитрий Шарков

ej \ 17-08

Прочитайте из первой строки число n , $n \leq 10$. Со второй строки - n чисел. Выведите в файл число, и сколько раз оно непрерывно употреблялось в цепочке.

Числа в диапазоне от -32 768 до 32 767.

input.txt**output.txt**

9
7 7 7 3 3 4 5 5 7

7-3
3-2
4-1
5-2
7-1

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 5 - 8 кл \ 7 - "Бомбардировка (упрощенная)"

dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2013 \ Day1 \ 1 - "Солдатская кухня"

Школьная алгебра

Автор: Дмитрий Шарков
ej \ 17-09

Трёхчлен $ax+by+cy$ от двух переменных x и y однозначно определяется коэффициентами a , b и c . Напишите программу, которая по заданным a , b и c выводит соответствующий трёхчлен, записанный с использованием алгебраических соглашений:

1. коэффициент при члене, содержащем переменную, опускается, если его модуль равен единице;
2. член, коэффициент при котором равен нулю, опускается (кроме случая, когда все коэффициенты равны нулю, тогда трёхчлен состоит из одной цифры 0);
3. знак "+" опускается, если он предшествует отрицательному коэффициенту;
4. знак "+" опускается, если он стоит в начале выражения (так называемый унарный плюс);
5. знак умножения между коэффициентом и переменной опускается.

При этом запрещено менять местами члены.

Формат ввода: во входном файле через пробел записаны целые коэффициенты a , b и c , каждый из которых не превосходит 30 000 по абсолютной величине.

Формат вывода: выходной файл должен содержать трёхчлен, записанный с использованием алгебраических соглашений.

Пример 1

input.txt

0 2 -1

output.txt

2x-y

Пример 2

input.txt

3 0 -2

output.txt

3-2y

18. Линейные массивы. Алгоритм поиска элемента. Метод флагов. Флаг печати (Линейные массивы. Алгоритм поиска элемента. Флаг печати разделителя)

Числовое зеркало

Автор: Евгений Пархоменко

[ej \ 18-01](#)

Прочитайте из первой строки файла ввода число n – количество элементов массива, $n \leq 10$, со второй строки – n элементов массива.

Выведите элементы массива в файл, расположив их в зеркальном порядке.

input.txt

```
6
2 7 3 4 1 5
```

output.txt

```
5 1 4 3 7 2
```

Совпадение чисел

Авторы: Адам Бардашевич, Дмитрий Шарков, Александр Птичкин

[ej \ 18-02](#)

Прочитайте из первой строки файла число n , $n \leq 10$, со второй строки – n чисел, с третьей строки – число x .

Определите, содержится ли число x среди чисел второй строки.

Выведите "Yes", если число x есть, или "No" – в противном случае.

input.txt

```
6
20 70 40 10 40 50
40
```

output.txt

```
Yes
```

[dl.gsu.by \ Программирование – начинающие \(P/O\)](#)

[1-10 \ 2 \ 1 - "Магазин"](#)

[dl.gsu.by \ Программирование – начинающие \(P/O\)](#)

[11-20 \ 1 \ 1 - "Машины"](#)

Первое вхождение

Автор: Антон Надёжкин

[ej \ 18-03](#)

Прочитайте из первой строки число n , $3 \leq n \leq 10$, со второй строки – n чисел, с третьей строки – число x .

Определите позицию первого вхождения x в цепочку чисел.

Выведите номер позиции числа x в цепочке, если x найдено, или "No" – в противном случае.

input.txt

```
6
20 70 40 40 10 50
40
```

output.txt

```
3
```

Последнее число

Автор: Дмитрий Шарков

[ej \ 18-04](#)

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n чисел, с третьей строки – число x .

Определите позицию последнего вхождения x в цепочку.

Выведите номер позиции числа x в цепочке, если x найдено, или "No" – в противном случае.

input.txt

```
6
20 70 40 10 40 50
40
```

output.txt

```
5
```

Количество вхождений

Авторы: Адам Бардашевич, Антон Надёжкин

[ej \ 18-05](#)

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n чисел, с третьей строки – число x .

Выведите количество вхождений числа x в цепочку чисел.

input.txt

```
6
20 70 40 10 40 50
40
```

output.txt

```
2
```

Интервал

Автор: Тимофей Абрамов

[ej \ 18-06](#)

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n элементов a_i , с третьей строки – числа x и y , $x \leq y$.

В первой строке файла выведите в порядке следования те элементы a_i , которые находятся на интервале $x..y$: $x \leq a_i \leq y$.

input.txt

```
6
```

output.txt

```
40 50 40
```

```
0 70 40 10 50 40
20 60
```

Количество элементов

Автор: Владислав Собин

[ej \ 18-07](#)

Прочитайте из первой строки n , $n \leq 10$, со второй строки – n элементов a_i , с третьей строки – x и y , $x \leq y$.

Выведите количество элементов, удовлетворяющих условию: $x \leq a_i \leq y$.

input.txt

```
6
0 70 40 10 50 40
20 60
```

output.txt

```
3
```

[dl.gsu.by \ Программирование – начинающие \(P/O\)](#)
[11-20 \ 6 \ 2 - "Класс"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Гом. обл. младшие \ 2002 \ День 2 \ 1 - "Числа Фибоначчи"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[GCSW \ 1999 \ Начинающие \ День 2 \ А - "Урок физкультуры"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Гомельская обл. \ 2011 \ День 1 \ 2 - "Реверсивная игра"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Гом. обл. ком. младшие \ 2002 \ 4 - "Новогодние подарки"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Гомельская обл. \ 2013 \ День 2 \ 1 - "Окна 1000"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Гомельская гор. \ 2001 \ Младшая группа \ 4 - "Палочки"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Гом. обл. ком. младшие \ 2001 \ 2 - "Задача о считалке"](#)

[ej](#)

2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Андрея Мищенко и Игоря Макеев \ В. Артём и дивизионы

[dl.gsu.by](#) \ Олимпиады по информатике

Российская (лич., Inet) \ 2008 \ Олимпиада_1 (11 октября 2008) \ 2 - "Генерация тестов"

[codeforces.ru](#) \ Тренировки

2011-2012 Командный чемпионат школьников Санкт-Петербурга по программированию (XIX) \ В. Икебана

[codeforces.ru](#) \ Тренировки

2012-2013 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов (ВКОШП 12, Интернет-тур) \ В. Реклама на заборе

[codeforces.ru](#) \ Архив

441A. Валера и антиквариат

[codeforces.ru](#) \ Архив

369B. Валера и олимпиада

[codeforces.ru](#) \ Архив

441C. Валера и трубы

Super Paintball

USACO October 2007 Gold division

Автор тестов: Сергей Кулик

[ej](#) \ 18-08

Коровы купили игру Супер-Пейнтбол. Бесси разделила пастбище на $N \times N$ квадратов ($1 \leq N \leq 100$) и составила список из K ($1 \leq K \leq 10\,000$) координат ($1 \leq R_i \leq N$; $1 \leq C_i \leq N$) каждого из ее противников в игре.

Пейнтбольное оружие может стрелять в одном из 8 направлений: север, юг, восток, запад, северо-восток, юго-восток, северо-запад, юго-запад.

Бесси хочет узнать количество способов выбрать свое положение так, чтобы она могла "одновременно поразить" всех своих противников. Она может поражать их также, если она находится в одном квадрате с противником.

Формат ввода: два разделенных пробелом целых числа N и K . Далее следует K строк. Строка i содержит R_i и C_i .

Формат вывода: одно целое число - количество положений Бесси, из которых она может поразить одновременно всех своих противников.

Пример 1**paint2.in**

```

4 3
2 1
2 3
4 1

```

paint2.out

```

5

```

Пример 2**paint2.in**

```

4 3
1 1
2 2
4 4

```

paint2. out

```

4

```

19. Сортировка подсчётом (Сортировка подсчётом. Методы ускорения операций ввода / вывода)

dl.gsu.by \ Олимпиады по информатике

GCSW \ 1999 \ Начинающие \ День 2 \ D - "Результаты стрельбы"

Неповторяющиеся числа

Автор: Владислав Собин

ej \ 19-01

Прочитайте из первой строки число n , $n \leq 100$, со второй строки – n чисел a_i , $0 \leq a_i \leq 10000$.

Перечислите в порядке возрастания и без повторений те числа, которые содержатся в исходной строке.

input.txt

```

7
2 7 4 1 4 5 2

```

output.txt

```

1
2
4
5
7

```

ej

2017-06-11 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Игоря Макеевко и Андрея Мищенко \ F. Игорь и удачные контесты

ej

2017-08-13 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ B. Одинаковые числа

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 1 \ 2 - "Агент 007"*

*dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 4 \ 2 - "Числа"*

Количество повторений

Автор: Евгений Пархоменко

ej \ 19-02

Прочитайте из первой строки число n , $n \leq 100$, со второй строки – n чисел a_i , $0 \leq a_i \leq 10000$.

Перечислите в порядке возрастания и без повторений те числа, которые содержатся в исходной строке. Дополнительно выведите информацию о том, сколько раз встречается число в исходной строке.

input.txt

```
7
2 7 4 1 4 5 2
```

output.txt

```
1, iterations: 1
2, iterations: 2
4, iterations: 2
5, iterations: 1
7, iterations: 1
```

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 3 \ 1 - "Знайка"*

Комментарий: полагайте, что числа во вводе неотрицательные

Литтлстроение

Открытое командное первенство УО МГПУ

Мозырь, 4 марта 2006 г.

ej \ 19-03

В стране Лиллипутии каждые десять лет проводится перепись населения. При заполнении анкеты каждый житель страны – литтл – должен, помимо всего прочего, указать также и свой возраст в годах (возраст своих несовершеннолетних детей взрослые литтлы заполняют сами).

По окончании переписи по всей стране устраиваются народные празднества и гуляния. Венцом же сего действия становится ежегодное литтлстроение, когда все граждане, участвовавшие в переписи, размещаются в одну длинную шеренгу, от самого младшего к самому старшему литтлу, а глава государства проходит вдоль шеренги с пирогом и даёт каждому литтлу откусить столько раз, сколько лет ему исполнилось. Считается, что эта церемония способствует единению нации.

Для контроля количества откусов от государственного пирога требуется сформировать соответствующий список, в котором указано, сколько откусов может сделать каждый литтл в порядке их следования в праздничной шеренге.

Формат ввода: в первой строке входного файла записано число n ($1 \leq n \leq 1\,000\,000$), - количество литтлов, принявших участие в переписи. Затем следуют n натуральных чисел из диапазона от 1 до 100 - перечень возрастов этих литтлов.

Формат вывода: выведите в выходной файл n строк, каждая из которых содержит одно число - допустимое количество откусов от государственного пирога, принявшего участие в переписи литтла в том порядке, в котором они будут расположены в праздничной шеренге.

input.txt

5
3 1 2 4 2

output.txt

1
2
2
3
4

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2005 \ Городская (3 декабря) \ Младшие \ 2 - "Вирус"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие, ком. \ 4 - "Задание на дом"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2000 \ Личные соревнования \ Начинающие \ День 2 \ В - "Построение"*

*codeforces.ru \ Архив
405A. Переключение гравитации*

Коллекционер

Автор: Ионас Пакальнишкис
ej \ 19-04

Мальчик Володя очень любит собирать коллекции, у него есть n марок разных видов. Для того, чтобы составить коллекцию, у него должно быть 10 марок одного вида в этой коллекции. Выведите количество коллекций, которые можно составить из этих марок.

Формат ввода:

Первая строка: число n . $1 \leq n \leq 100\,000$.
Вторая строка: n чисел $a[i]$ – виды марок. $1 \leq a[i] \leq 100$.

Формат вывода: искомое число.

input.txt

20
2 2 2 2 2 3 2 3 2 3 2 3 2 2 3 3 3 3 3

output.txt

2

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Начинающие \ День 2 \ 1 - "Заболевший футболист"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2015 \ День 1 \ 1 - "День именинника"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2001 \ День 2 \ 4 - "Урок физкультуры"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинающие, ком. \ 4 - "Сила тока"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 5 - "Красивые суммы"*

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2010 \ December \ Bronze \ 2 - "Bad Random Numbers"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2011 \ День 2 \ 2 - "Чемпионат по футболу"*

*codeforces.ru \ Архив
432B. Футбольная форма*

*ej
2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Андрея
Мищенко и Игоря Макеенко \ С. Байтландское путешествие*

Бараболия

*Авторы: Илья Смоляр, Игорь Макеенко, Андрей Мищенко
ej \ 19-05*

В Байтландии набирает популярность новая карточная игра "Бараболия". Григорий хочет стать чемпионом в этой игре.

Правила игры "Бараболия" следующие:

- один из соперников берёт себе n карточек из общей колоды, а другой (Григорий), берёт m карточек ($0 \leq n, m \leq 100\,000$);
- Григорий ходит первым (и единственным);
- Григорий может забрать все карточки соперника $a[i]$, для которых в наборе карточек Григория есть карточки $b[j]$ с таким же значением ($0 \leq a[i], b[j] \leq 10^6$);

Помогите Григорию узнать, какое максимальное количество карточек он сможет забрать у соперника.

Формат ввода:

n
 $a[1] a[2] \dots a[n]$
 m
 $b[1] b[2] \dots b[m]$

Формат вывода:

ans ответ на задачу

gamewi.n

10
 1 2 3 4 5 6 1 8 1 10
 5
 1 2 3 4 2

wingam.e

6

20. Поиск наибольшего элемента. Константы $maxint$, $maxlongint$ (Поиск наибольшего элемента. Функции max , min)

dl.gsu.by \ Программирование – начинающие (P/O)
 11-20 \ 4 \ 2 - "Белоснежка"

dl.gsu.by \ Олимпиады по информатике
 Гом. обл. младшие \ 2003 \ День 2 \ 2 - "Оценки"

dl.gsu.by \ Олимпиады по информатике
 Гом. обл. младшие \ 2001 \ День 1 \ 1 - "Деревья"

dl.gsu.by \ Олимпиады по информатике
 Гом. обл. младшие \ 2001 \ День 1 \ 2 - "Бетонные блоки"

dl.gsu.by \ Олимпиады по информатике
 GCSW \ 2002 \ Начинающие, ком. \ 1 - "Соревнования"

Наибольший элемент

Автор: Владислав Собин

ej \ 20-01

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n элементов.

Выведите в файл наибольший элемент.

input.txt

6
2 7 4 1 4 5

output.txt

7

dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 2 \ 2 - "Санта Клаус"

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 4 \ 2 - "Конкурс"

dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 3 \ 1 - "От планеты до планеты"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 1999 \ Начинаящие \ День 1 \ D - "Мотки проволоки"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинаящие, ком. \ 6 - "Мистер пузырь"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2000 \ Личные соревнования \ Начинаящие \ День 1 \ B - "Сбор макулатуры"

Индекс

Автор: Тимофей Абрамов

ej \ 20-02

Прочитайте из первой строки n , $n \leq 10$, со второй строки – n элементов.

Выведите в файл индекс наибольшего элемента и его значение согласно формату примера вывода.

Если в строке несколько наибольших элементов, выведите индекс первого из них.

input.txt

6
2 7 4 1 7 5

output.txt

a[2]=7

dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 1 \ 2 - "Самый большой торт"

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 2 \ 2 - "Вклад"

dl.gsu.by \ Программирование – начинающие (Р/О)
21-30 \ 3 \ 2 - "Соседние числа максимума"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2015 \ Городская 1-11 кл, 14 ноября \ 9 - 11 кл \ 1 - "Условно минимальный"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Начинающие \ День 2 \ 2 - "Бег на 100 метров"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 2 \ 2 - "Урок физкультуры"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 1 \ 1 - "Лунатики"

Надоедливый сынок

Автор: Ионас Пакальнишкис

ej \ 20-03

Жил-был маленький мальчик Петя, он очень любил математику, поэтому он часто у своих родных просил разные задачки на математику. И вот, его мама решила дать надоедливому сыночку задачу, и сказала: "Если ты решишь её, то я тебе разрешу поиграть на компьютере, а если не решишь, то я тебя лишу карманных на месяц." Помогите Пете решить эту задачу.

Вот суть маминой задачи: дано n чисел, нужно найти разность максимального и минимального из этих чисел.

Формат ввода:

Первая строка: число n . $1 \leq n \leq 400\,000$.

Вторая строка: n чисел $a[i]$. $-1\,000\,000 \leq a[i] \leq 1\,000\,000$.

Формат вывода: искомое число.

input.txt

3
13 15 17

output.txt

4

dl.gsu.by \ Олимпиады по информатике
GCSW \ 1999 \ Начинающие, ком. \ В - "Рубашки"
Комментарий: полагайте, что $D_i \leq 20$

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2005 \ Городская (3 декабря) \ Младшие \ 1 - "Дятлик школьник"

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2003 \ День 1 \ 2 - "Грибники-2"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2000 \ Командные соревнования \ Командный тур (начинающие) \ В - "Школьная олимпиада"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2001 \ День 2 \ 6 - "Задача о росте"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2010 \ День 1 \ 1 - "Детский сад"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2011 \ Школьная 1-8 кл, 20 марта \ 7 - 8 кл \ 1 - "Min+Max"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-11 кл, 9 ноября \ 9 - 11 кл \ 1 - "Расстояния в массиве"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2003 \ День 2 \ 1 - "Самый высокий"*

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 7 \ 2 - "Выговор"*

Смещающиеся числа

Автор: Адам Бардашевич

ej \ 20-04

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n элементов массива. Абсолютное значение каждого элемента не превышает $maxint$.

Найдите в массиве первый по порядку наибольший и первый по порядку наименьший элементы, поменяйте их значения местами.

Полученный массив выведите в файл.

Ввод и вывод данных должен проводиться так, как показано в примере фрагмента программы.

Пример фрагмента программы:

```
...  
  for i:=1 to n do read(a[i]);  
...
```

```
for i:=1 to n-1 do write(a[i], ' ');  
writeln(a[n]);  
...
```

input.txt

```
6  
2 7 4 1 7 5
```

output.txt

```
2 1 4 7 7 5
```

codeforces.ru \ Тренировки

[2013-2014 Цикл интернет-олимпиад. Третья командная олимпиада, базовый уровень \(19 октября 2013 года\) \ В. Свидание](#)

[dl.gsu.by](#) \ Олимпиады по информатике

[Гомельская гор. \ 2017 \ Городская 1-9 кл, 7 апреля \ 5-9 кл \ 7 - "Команды \(упрощённая\)"](#)

[dl.gsu.by](#) \ Олимпиады по информатике

[Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 5 - 8 кл \ 6 - "Конкуренты - 2"](#)

[dl.gsu.by](#) \ Олимпиады по информатике

[Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 9 - 11 кл \ 1 - "Веселые Массивы"](#)

[dl.gsu.by](#) \ Олимпиады по информатике

[Гомельская обл. \ 2012 \ День 2 \ 1 - "Утренник"](#)

[dl.gsu.by](#) \ Программирование – начинающие (Р/О)

[11-20 \ 4 \ 4 - "Конкурс на внимательность"](#)

Юзик

Автор: Антон Надёжкин

[ej \ 20-05](#)

Акционер Юзик знает, сколько раз изменялись в стоимости его акции – n раз. Но вот математику он не знает. Помогите Юзику узнать максимальную стоимость акций за некоторый период времени.

Формат ввода: во входном файле задано количество величин n ($0 < n \leq 10$) и сами величины, на которые изменялись акции. Они являются целыми, лежат в интервале $-30\,000..30\,000$. Начальная стоимость акций равна нулю.

Формат вывода: в выходном файле размещается максимальная стоимость акций за указанный период.

input.txt

```
8  
1 -2 1 1 1 -2 1 -1
```

output.txt

```
2
```

Одинаковоподрядирующие

Автор: Алексей Ропан

ej \ 20-06

Прочитайте из первой строки n , $n \leq 10$, со второй строки – n чисел.

Выведите в файл максимальную длину непрерывной цепочки одинаковых чисел.

input.txt

8
7 7 1 1 1 5 4 4

output.txt

3

dl.gsu.by \ Олимпиады по информатике

Гом. обл. младшие \ 2002 \ День 1 \ 1 - "Олимпиада"

dl.gsu.by \ Программирование – начинающие (P/O)

11-20 \ 8 \ 2 - "Олимпиада по математике"

First Maximum

Авторы: Александр Кулицкий, Алексей Ропан, Игорь Прищепо

ej \ 20-07

Прочитайте из первой строки число n , $1 \leq n \leq 10$, со второй строки – n чисел.

Выведите в файл первую максимальную по длине непрерывную цепочку одинаковых чисел.

input.txt

7
2 7 7 1 1 1 3

output.txt

1 1 1

Футбольный чемпионат

Автор: Алексей Ропан

ej \ 20-08

На чемпионате по футболу получилось так, что все команды с номерами от 1 до n набрали одинаковое количество очков. По футбольным правилам места у команд должны быть распределены согласно количеству забитых и пропущенных ими мячей. Приоритет в распределении мест имеет количество забитых мячей.

Надо помочь судьям узнать, кто является победителем. Гарантируется, что победитель будет один.

Формат ввода:

n количество команд ($2 \leq n \leq 100$)
 $z_1 p_1$ описания команд в порядке возрастания их номеров
 $z_2 p_2$ z - количество забитых мячей
 ... p - количество пропущенных мячей

$$z_n p_n \quad (0 \leq z, p \leq 100)$$

Формат вывода: номер команды-победительницы и её результат.

champ.in

```
3
2 3
14 10
14 9
```

champ.out

```
Team number: 3
Win:Loss - 14:9
```

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 2 - "Строка с максимальной суммой"

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2003 \ Начинающие, ком. \ 1 - "Наказание"

dl.gsu.by \ Олимпиады по информатике

GCSW \ 1999 \ Начинающие \ День 2 \ Е - "Разгрузка арбузов"

ej

2017-06-11 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Игоря Макеевко и Андрея Мищенко \ В. Илья и пицца

Селения

Областной конкурс по программированию

Гомель

ej \ 20-09

Имеется k селений, $k \leq 10$. Если в селении i расположить пункт скорой помощи, то поездка по вызову в селение j займёт время $a_{i,i} + a_{i,j}$ ($1 \leq i, j \leq k, i \neq j$).

Найдите номер селения i , от которого поездка в самое удалённое (по времени) селение занимало бы минимальное время. Если таких селений несколько, выведите селение с наименьшим номером.

Все $a_{i,j} > 0$, элемент $a_{i,j}$ может быть не равен элементу $a_{j,i}$. Максимальная длина поездки $< \max int$.

Формат ввода:

k

$a_{1,1} \ a_{1,2} \ \dots \ a_{1,k}$

$a_{2,1} \ a_{2,2} \ \dots \ a_{2,k}$

...

$a_{k,1} \ a_{k,2} \ \dots \ a_{k,k}$

Формат вывода: номер оптимального селения.

Комментарий: времена поездки в самые удалённые селения для примера ввода:

1: 1+10=11
2: 5+4=9
3: 7+6=13
4: 1+13=14

Минимальное время поездки в самое удалённое селение находится во второй строке. Таким образом, результат - селение 2.

input.txt

```
4
1 10 7 3
2 5 4 1
6 3 7 1
5 11 13 1
```

output.txt

```
2
```

[ej](#)

2018-05-27 Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ F. Футбол

[ej](#)

2017-07-09 [Дорешивание] Тренировочная олимпиада КЮП. Контест Вячеслава Козловского и Ивана Ревта \ В. Дима на сессии

[ej](#)

2017-07-09 [Дорешивание] Тренировочная олимпиада КЮП. Контест Вячеслава Козловского и Ивана Ревта \ G. Жизненная задача

21. Обмен значений двух переменных. Циклический сдвиг элементов массива

[ej](#)

2017-06-18 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ D. Смена элементов

Глобальный сдвиг

Автор: Егор Казаченко

[ej \ 21-01](#)

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n элементов массива.

Сдвиньте элементы массива на одну позицию влево так, чтобы в $a[1]$ было помещено значение $a[2]$, в $a[2]$ – значение $a[3]$ и так далее. В $a[n]$ поместите значение $a[1]$.

В программе можно использовать не более одного массива.

Ввод и вывод данных должен проводиться так, как показано в примере фрагмента программы. В программе нельзя использовать команду *swap*.

Пример фрагмента программы:

```
...  
for i:=1 to n do read(a[i]);  
...  
for i:=1 to n-1 do write(a[i], ' ');  
writeln(a[n]);  
...
```

input.txt

```
6  
1 10 2 20 3 30
```

output.txt

```
10 2 20 3 30 1
```

Глобальный сдвиг 2

Автор: Руслан Кулицкий

[ej \ 21-02](#)

Прочитайте из первой строки число n , $n \leq 30$, со второй строки – n элементов массива.

Сдвиньте элементы массива на одну позицию вправо так, чтобы в $a[1]$ было помещено значение $a[n]$, в $a[2]$ – значение $a[1]$ и так далее. В $a[n]$ поместите значение $a[n-1]$.

В программе можно использовать не более одного массива.

Ввод и вывод данных должен проводиться так, как показано в примере фрагмента программы.

В программе нельзя использовать команду *swap*.

Пример фрагмента программы:

```
...  
for i:=1 to n do read(a[i]);  
...  
for i:=1 to n-1 do write(a[i], ' ');  
writeln(a[n]);  
...
```

input.txt

```
6  
1 10 2 20 3 30
```

output.txt

```
30 1 10 2 20 3
```

Зеркальный переворот

Автор: Сергей Кулик

[ej \ 21-03](#)

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n элементов массива.

Поменяйте местами элементы массива зеркально относительно центра так, как показано в примере ввода-вывода.

В программе можно использовать не более одного массива.

В программе нельзя использовать команду *reverse*.

Ввод и вывод данных должен проводиться так, как показано в примере фрагмента программы.

Пример фрагмента программы:

```
...  
for i:=1 to n do read(a[i]);  
...  
for i:=1 to n-1 do write(a[i], ' ');  
writeln(a[n]);  
...
```

input.txt

```
5  
1 10 2 20 3
```

output.txt

```
3 20 2 10 1
```

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2000 \ Младшая группа \ 1 - "Маршрут туристов"

[ej](#)

2017-06-11 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Игоря Макеенко и Андрея Мищенко \ D. Склад

22. Сортировка выбором. Сумма членов арифметической прогрессии

Неубывающая последовательность

Автор: Георгий Коноплич

[ej \ 22-01](#)

Прочитайте из первой строки число n , $n \leq 10$, со второй строки – n элементов массива.

Отсортируйте элементы массива по неубыванию и выведите их в файл.

input.txt

```
6  
2 7 4 1 4 5
```

output.txt

```
1 2 4 4 5 7
```

Гигант и камни

Авторы: Виталий Гринич, Максим Веретенников

[ej \ 22-02](#)

Однажды гиганту Павлику дали задание. Ему надо n камней разложить так, чтобы они лежали по размеру от меньшего к большему. Он отсортировал камни сортировкой выбором. Выведите, за сколько операций обмена он это сделал, и что он получил в итоге.

Формат ввода:

n количество камней ($0 \leq n < 50$)
 $a[i]$ размер камней ($0 < a[i] < 150$) – целые числа

Формат вывода:

k количество перестановок
 $b[i]$ конечный результат

pavlik.in

5
 2 3 1 4 5

pavlik.out

2
 1 2 3 4 5

dl.gsu.by \ Программирование – начинающие (P/O)

11-20 \ 5 \ 2 - "Писатель"

Комментарий: номера страниц могут быть больше N

dl.gsu.by \ Олимпиады по информатике

GCSW \ 1999 \ Начинающие \ День 2 \ C - "Нефтедобыча"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 5 - 7 кл \ 6 - "Две суммы"

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2002 \ Начинающие \ День 1 \ 1 - "Билли"

Объединённая последовательность

Автор: Александр Птичкин

[ej \ 22-03](#)

Прочитайте из первой строки an , $an \leq 10$, со второй строки – an чисел последовательности a .

Прочитайте из третьей строки bn , $bn \leq 10$, с четвёртой строки – bn чисел последовательности b .

Выведите в файл объединённую последовательность чисел из a и b , отсортированную по неубыванию.

В программе можно использовать не более одного массива.

input.txt

4
 2 7 1 3
 5
 3 5 3 2 4

output.txt

1 2 2 3 3 3 4 5 7

codeforces.ru \ Архив

456A. Ноутбуки

Урок физкультуры

Автор: Сергей Кулик

ej \ 22-04

На уроке физкультуры учителю надо расставить учеников в порядке возрастания роста. Если их рост равен, то вперёд становится ученик, сила которого больше. Помогите учителю расставить учеников.

Гарантируется, что сила у всех учеников разная.

Формат ввода:

n количество учеников ($0 < n < 151$)

$a_1 b_1$ $a_i b_i$ - рост и сила i ученика соответственно ($0 < a_i, b_i < 1000001$)

$a_2 b_2$

...

$a_n b_n$

Формат вывода: выведите рост и силу каждого ученика после расстановки.

lesson.in

4
4 3
1 2
7 8
5 6

lesson.out

1 2
4 3
5 6
7 8

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 5 - 8 кл \ 6 - "Борцы сумо"

Сортировка чётных и нечётных

Городская олимпиада по информатике

Мозырь, ноябрь 1998 г.

ej \ 22-05

Вам даётся массив, который нужно отсортировать следующим образом:

- отсортировать по неубыванию отдельно элементы, стоящие на чётных позициях;
- отсортировать по неубыванию отдельно элементы, стоящие на нечётных позициях.

Формат ввода: в первой строке содержится n , $0 < n \leq 10$ – количество элементов в массиве. В последующих n строках содержатся элементы массива, по одному в строке.

Формат вывода: вам нужно вывести отсортированный массив так, чтобы в каждой строке находилось ровно одно число.

Комментарий: в программе можно использовать не больше одного массива.

input.txt

5
10
9
-1
5
3

output.txt

-1
5
3
9
10

[codeforces.ru \ Архив](https://codeforces.ru/archive)

[802М. Первоапрельская задача \(простая\)](#)

[ej](#)

[2017-06-18 \[Дорешивание\] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ В. Владик и футболки](#)

Муромские мужики

Районная олимпиада по информатике

Гомель, 5 декабря 1999 г.

[ej \ 22-06](#)

На торговую дорогу между Муромом и Новгородом Соловей-Разбойник положил огромный камень весом n пудов, $0 < n < 32000$.

Камень закрывает дорогу и мешает проезду купцов с товарами из Мурома в Новгород и обратно. Муромские мужики решили убрать камень с дороги.

Всего в Муроме есть k человек, $0 < k \leq 10$, каждый из которых может поднять m_i пудов, $0 < m_i < 32000$, где i – номер человека.

Требуется выяснить, каково минимальное количество человек, которое потребуется, чтобы убрать камень с дороги.

Если муромских мужиков недостаточно, чтобы убрать камень, следует вывести число 0.

Формат ввода:

n k

m_1 m_2 ... m_k

Формат вывода:

t

Здесь t – минимальное количество человек, которое потребуется, чтобы убрать камень.

input.txt

100 5
48 3 5 32 18

output.txt

4

dl.gsu.by \ Олимпиады по информатике

Гомельская обл. \ 2009 \ День 2 \ 1 - "Азартный Шрэк "

Ограбление

Авторы: Игорь Прищепо, Алексей Ропан

[ej \ 22-07](#)

Вовочка решил ограбить машину инкассации. Он разработал идеальный план ограбления.

Из разных источников Вовочка получил информацию о расположениях машины в разные моменты времени. Он хочет узнать путь следования машины.

Формат ввода:

n количество источников ($1 \leq n \leq 100$)
 $k_1 h_1 m_1$ k_i - номер пункта, $h_i m_i$ - время: часы и минуты
 $k_2 h_2 m_2$ ($0 \leq k_i \leq 255$, $0 \leq h_i \leq 23$, $0 \leq m_i \leq 59$)
 ...
 $k_n h_n m_n$

Формат вывода: выведите порядок следования машины в формате "hh:mm - k". hh:mm - время (часы и минуты соответственно), k - номер пункта в данное время.

time.in

5
 233 10 30
 173 09 01
 44 11 10
 252 12 00
 139 08 30

time.out

08:30 - 139
 09:01 - 173
 10:30 - 233
 11:10 - 44
 12:00 - 252

[codeforces.ru \ Архив](#)

[710B. Оптимальная точка на прямой](#)

[ej](#)

2017-08-13 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ F. Необычная сортировка

[codeforces.ru \ Архив](#)

[1174B. Ихаб и странный человек](#)

[codeforces.ru \ Архив](#)

[388A. Лиса и коробки](#)

[codeforces.ru \ Архив](#)

[384B. Мультизадачность](#)

[codeforces.ru \ Архив](https://codeforces.ru/archive)

[389A. Лиса и игра с числами](#)

[ej](#)

2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Андрея Мищенко и Игоря Макеенко \ D. Гриша и автоугон

23. Поиск последовательностей в массиве. Метод сканирующей прямой. Сложность алгоритма

Добрыня Микитич и Баба Яга

Авторы: Владислав Микитич, Алексей Ропан, Сергей Коноплич, Владимир Миранович

[ej \ 23-01](#)

После утомительного сражения со Змеем Горынычем Добрыня случайным образом набрёл на избушку Бабы Яги. Микитич показал Яге карту с маяками и спросил, как добраться коротким путём домой. Яга же, узнав о смерти Горыныча (который так и не возвратил ей долг), решила отомстить Добрыне и дала неправильную информацию о номерах маяков, по которым он должен дойти до дома.

После ещё более утомительного хождения по лесу Микитич снова вернулся к избушке. Но, на этот раз, Баба Яга испугалась, и не обманула - дала номера маяков, такие, что если выписать номера новых маяков, которые совпадают со старыми, то Добрыня дойдёт по этому пути домой. Помогите Микитичу в этой проблеме.

Формат ввода:

n	количество старых маяков ($0 < n < 11$)
$a_1 a_2 \dots a_n$	номера старых маяков ($0 < a_i < 1000001$)
m	количество новых маяков ($0 < m < 11$)
$b_1 b_2 \dots b_m$	номера новых маяков ($0 < b_i < 1000001$)

Формат вывода: номера совпадающих маяков. Порядок вывода совпадает с порядком ввода старых маяков.

mikitich.in

```
8
1 2 3 2 2 3 4 1
6
2 3 1 3 3 5
```

mikitich.out

```
1 2 3 3
```

Содержимое

Миниолимпиада КЮП

Мозырь, 8 декабря 1996 г.

[ej \ 23-02](#)

Даны массивы $a[m]$, и $b[n]$, $n < m \leq 10$. Выясните, содержится ли массив b в массиве a .

Массив b следует считать содержащимся в a , если в a существует непрерывная последовательность элементов, совпадающая с массивом b .

Формат ввода: две пары строк с информацией о массивах a и b . В каждой паре первая строка – размер соответствующего массива, вторая – его элементы.

Формат вывода: строка "B is contained in A" или строка "B is not contained in A".

input.txt

```
7
1 7 13 5 6 8 4
4
13 5 6 8
```

output.txt

```
B is contained in A
```

Большее из частейших

Автор: Олег Любинский

Авторы тестов: Олег Любинский, Сергей Коноплич, Адам Бардашевич, Сергей Кулик

[ej \ 23-03](#)

Прочитайте из первой строки число n , $1 \leq n \leq 15\,000$, со второй строки – n элементов массива $a[i]$, $-32\,767 \leq a[i] \leq 32\,767$.

Выведите в файл то число, которое встречается в массиве наибольшее количество раз. Если таких чисел несколько, выведите большее из них.

input.txt

```
7
2 7 4 1 4 5 2
```

output.txt

```
4
```

[dl.gsu.by \ Программирование – начинающие \(P/O\)](#)

[11-20 \ 7 \ 1 - "Среднее арифметическое"](#)

Комментарий: если подходящих чисел несколько, выведите номера двух последних из них

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская обл. \ 2014 \ День 1 \ 1 - "Футбол"](#)

Максимум

Городская олимпиада по программированию, КЮП

Мозырь, 4 марта 2000 г.

Соавтор тестов: Владимир Миранович

[ej \ 23-04](#)

В наборе чисел найдите непрерывную неубывающую последовательность чисел максимальной длины. Если подходящих последовательностей несколько, выведите первую из них. $-maxint \leq a[i] \leq maxint$.

Формат ввода: первая строка: $n \leq 300\,000$. Вторая строка: элементы набора чисел.

Формат вывода: искомая последовательность.

input.txt

```
7
5 1 2 3 4 3 7
```

output.txt

```
1 2 3 4
```

Перевёртыши

Городская олимпиада по программированию, КЮП

Мозырь, 4 марта 2000 г.

Соавторы тестов: Владислав Собин, Сергей Кулик

[ej \ 23-05](#)

Задан числовой массив $a[n]$. Найдите отрезок массива максимальной длины, в котором первое число равно последнему, второе – предпоследнему и так далее.

Если подходящих отрезков несколько, выведите первый по порядку.

Формат ввода: первая строка: n . $n \leq 7000$. Вторая строка: элементы массива. $-maxint \leq a[i] \leq maxint$

Формат вывода: искомый отрезок.

input.txt

```
9
5 3 4 7 4 3 6 6 3
```

output.txt

```
3 4 7 4 3
```

[dl.gsu.by \ Олимпиады по информатике](#)

[Белорусская \ 2014 \ День 1 \ 2 - "Шведский стол"](#)

[codeforces.ru \ Архив](#)

[427B. Перевод заключенных](#)

[codeforces.ru \ Архив](#)

[465B. Входящие \(100500\)](#)

[codeforces.ru \ Архив](#)

[387B. Георгий и раунд](#)

[codeforces.ru \ Тренировки](#)

[2012-2013 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов \(ВКОШП 12, Интернет-тур\) \ D. Лягушонок Билли](#)

Разрушения

Автор: Иван Ревт

ej \ 23-06

Ровно год прошел со времени вступления землян в контакт с инопланетной цивилизацией. И все это время они производили точечные удары по районам Земли, а оборона землян была парализована электромагнитным импульсом.

Карта Земли представлена матрицей из n строк и m столбцов, где каждая клетка матрицы - район Земли. За время войны лучшие умы человечества выяснили, что удары наносятся только по клеткам, у которых сумма координат является степенью двойки.

Главы стран мира не могут предотвратить разрушения и обратились к вам за помощью: узнайте, сколько процентов поверхности Земли будет уничтожено.

Формат ввода:

В единственной строке записаны два целых числа n и m ($0 \leq n, m \leq 10^6$) - размеры карты.

Формат вывода:

Выведите единственное вещественное число с 7 знаками после точки - процент уничтоженных районов Земли от общего количества районов.

Примечание: степень двойки – это число в форме 2^k , например: 1, 2, 4, 8...

two.in

4 4

two.out

31.2500000

Музей

Городская олимпиада по программированию, КЮП

Мозырь, 4 марта 2000 г.

ej \ 23-07

В одной из комнат музея установлена электронная система, которая регистрирует время прихода и ухода каждого посетителя.

Напишите программу, определяющую непрерывный интервал времени в течение дня, когда количество одинаковых посетителей в комнате было максимальным. Если таких интервалов несколько – выведите первый по порядку течения времени.

Формат ввода: первая строка содержит количество человек n , посетивших музей в течение дня, $0 < n \leq 10$. Остальные строки имеют вид Ч1 М1 Ч2 М2. Здесь Ч1 М1 – время прихода очередного посетителя в музей, Ч2 М2 – время его ухода.

Формат вывода: первая строка: Ч1 М1 Ч2 М2 – начало и конец искомого интервала времени ненулевой длины. Предваряйте количество минут лидирующими нулями, если это необходимо. Вторая строка: количество посетителей, находившихся в музее в течение этого интервала.

input.txt

4

08 30 12 00

09 40 13 20

09 50 13 00

15 20 17 30

output.txt

9 50 12 00

3

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2016 \ День 2 \ 1 - "День Святого Байта"

informatics.mccme.ru
Задача №112772. Ремонт асфальта

codeforces.ru \ Архив
673С. Медвежонок и цвета

[ej](#)
2017-06-11 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Игоря
Макеев и Андрея Мищенко \ Е. Игорь и взломы

codeforces.ru \ Архив
1163В2. Кошачья вечеринка (усложнённая версия)

24. Тип `char`. Коды символов. Символ одинарной кавычки. Функции `chr`, `ord`, `pred`, `succ`. Тип `char` в `inc`, `dec`, `for`. Функция `uppercase`

i-тый символ

Автор: Максим Веремеев
[ej \ 24-01](#)

Прочитайте с первой строки файла ввода число n , $1 \leq n \leq 500$, со второй строки – символ с номером n , и выведите его в файл вывода.

input.txt

3
abcd

output.txt

c

Chr

Автор: Филипп Головацкий
[ej \ 24-02](#)

Прочитайте из файла ввода число x , выведите в файл вывода символ с кодом x в формате, приведённом в примере.

input.txt

65

output.txt

Symbol: 'A'

ASCII

Автор: Егор Липский

ej \ 24-03

Прочитайте числа $k1$ и $k2$ из файла ввода.

Поместите в файл вывода информацию о таблице кодов ASCII на промежутке $k1..k2$ в формате, приведённом в примере вывода.

input.txt

65 70

output.txt

#65='A'

#66='B'

#67='C'

#68='D'

#69='E'

#70='F'

Chr u Ord

Автор: Владислав Собин

ej \ 24-04

В файле ввода может находиться строка вида " $x=код$ ", где *код* – это код произвольного символа, например " $x=65$ ", либо вида " $c='символ'$ ", где *символ* – это произвольный символ, например " $c='A'$ ".

В случае, когда введённая строка имеет вид " $x=код$ ", поместите в файл вывода символ с кодом x в формате, который приведён в примере вывода.

Когда введённая строка имеет вид " $c='символ'$ ", поместите в файл вывода код символа c .

Пример 1

input.txt

$x=65$

output.txt

Symbol: 'A'

Пример 2

input.txt

$c='A'$

output.txt

Code: 65

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 5 - 8 кл \ 5 - "Номер буквы"

dl.gsu.by \ Олимпиады по информатике

Гом. обл. младшие \ 2001 \ День 1 \ 5 - "Задача о шахматах"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 5 - 8 кл \ 7 - "Спаси робота (упрощённая)"

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская гор. \ 2016 \ Школьная 1-11 кл, 5 октября \ 9-11 кл \ 7 - "Общежитие \(упрощённая\)"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 5 - 7 кл \ 7 - "Матрица \(упрощенная\)"](#)

[codeforces.ru \ Архив](#)

[710A. Ходы короля](#)

[codeforces.ru \ Архив](#)

[707A. Фотографии Брейна](#)

[codeforces.ru \ Архив](#)

[387A. Георгий и сон](#)

[codeforces.ru \ Архив](#)

[445A. DZY любит шахматную доску](#)

Перевёртыши 2

Автор: Сергей Кулик

[ej \ 24-05](#)

Задано n маленьких букв латинского алфавита. Выведите длину максимального из перевертышей, который можно составить из этих букв.

Формат ввода: первая строка: n ($n < 500001$). Вторая строка: n букв.

Формат вывода: длина максимального из перевертышей, который можно составить из заданных букв.

input.txt

5
aazbc

output.txt

3

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская гор. \ 2015 \ Школьная 1-11 кл, 7 октября \ 9 - 11 кл \ 5 - "Библиотека"](#)

[ej](#)

[2017-06-18 \[Дорешивание\] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ Е. Больше или меньше?](#)

[codeforces.ru \ Тренировки](#)

[2010-2011 командный чемпионат школьников Санкт-Петербурга по программированию \(XVIII\) \ В. Шахматная головоломка](#)

25. Тип string. Пустая строка. Сложение строк. Символы в строковой переменной. Процедура setlength. Функции length и copy. Функция eof. Типы shortstring и ansistring

Строчка

Автор: Руслан Кулицкий

[ej \ 25-01](#)

Прочитайте слово из первой строки файла, распечатайте все символы этого слова в столбец.

input.txt

Test

output.txt

T
e
s
t

Интересные строки

Автор: Иван Киринский

[ej \ 25-02](#)

Из двух данных слов одинаковой длины составьте новое слово следующим образом: берётся одна буква из первого слова, затем – из второго, затем – опять из первого и так далее. Длина слов не превышает 255 символов.

input.txt

Psa
acl

output.txt

Pascal

[ej](#)

[2018-05-27 Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ D. Есть ли палиндром?](#)

[ej](#)

[2017-09-03 \[Дорешивание\] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ D. GG!](#)

[ej](#)

[2017-08-13 \[Дорешивание\] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ C. DeNiS](#)

*dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 5 \ 4 - "Вирус"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2015 \ День 2 \ 1 - "Краткость - сестра таланта"*

Замена

Автор: Алексей Галай

ej \ 25-03

В слове замените указанную букву на другую, также указанную.

Первая строка файла ввода: исходное слово. Вторая строка: искомая буква. Третья строка: буква для замены.

В программе можно использовать не более одной переменной типа string.

input.txt

abcdab
a
b

output.txt

bbcdbb

ej

*2017-07-09 [Дорешивание] Тренировочная олимпиада КЮП. Контекст Вячеслава Козловского
и Ивана Реева \ А. Валера и экзамен*

Перевод строки

Авторы: Сергей Кулик, Дмитрий Шарков

ej \ 25-04

Прочитайте слово s из первой строки файла, число n – со второй. Разделите s на две части и выведите их в файл. n – количество символов во второй части s .

В программе нельзя использовать операторы цикла.

input.txt

tourist
5

output.txt

to
urist

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 4 - "Лишние буквы"*

Странные слова

Автор: Тимофей Абрамов

[ej \ 25-05](#)

Петя и Вася играли в игру под названием "Странные слова". По правилам игры вам даётся слово и какая-либо буква из него. Необходимо назвать часть этого слова, находящуюся между первым и последним вхождением этой буквы. Вася решил сжульничать, и просит вас напечатать программу для нахождения части слова.

input.txt

```
crocodile  
с
```

output.txt

```
сroc
```

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2005 \ January \ BRONZE \ 1 - "BANKBAL"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 1 - 4 кл \ 18 - "Задача 18"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие, ком. \ 3 - "Точки"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2001 \ Младшая группа \ 5 - "Вирус"*

Палиндром

Автор: Сергей Коноплич

[ej \ 25-06](#)

Палиндром – это слово, которое одинаково читается как слева направо, так и справа налево.

Прочитайте из файла слово и выведите в файл один из результатов: "Palindrome", "Non palindrome".

input.txt

```
ABBA
```

output.txt

```
Palindrome
```

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 2 \ 4 - "Шифр"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинающие \ День 1 \ 2 - "Рейтинг сайтов"*

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская гор. \ 2005 \ Городская \(3 декабря\) \ Младшие \ 4 - "Дятловский телефон"](#)

Баланс скобок

[ej \ 25-07](#)

В файле ввода находится строка, состоящая из открывающих и закрывающих круглых скобок.

Скобки называются сбалансированными, если для каждой открывающей скобки найдётся парная ей закрывающая, стоящая справа от неё, а для каждой закрывающей – парная ей открывающая, стоящая слева от неё.

Определите, сбалансированы ли скобки, и выведите одно из сообщений: "Balanced", "Not balanced".

input.txt

(())

output.txt

Balanced

[dl.gsu.by \ Олимпиады по информатике](#)

[Гом. обл. младшие \ 2003 \ День 2 \ 3 - "Таблица"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[Гом. обл. ком. младшие \ 2002 \ 5 - "Компрессия строк"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[Гом. обл. младшие \ 2002 \ День 1 \ 3 - "Слова-Х"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская гор. \ 2002 \ Младшая группа \ 3 - "Programming"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[Московская \ 2012 \ А - "Предложения с большой буквы"](#)

[codeforces.ru \ Архив](#)

[802G. Ложные Новости \(простая\)](#)

[ej](#)

[2017-07-09 \[Дорешивание\] Тренировочная олимпиада КЮП. Контекст Вячеслава Козловского и Ивана Ревта \ Е. Клещ в ловушке](#)

[codeforces.ru \ Архив](#)

[431A. Черный квадрат](#)

[codeforces.ru \ Архив](#)
[424А. Паша и приседания](#)

[codeforces.ru \ Архив](#)
[405В. Эффект домино](#)

[codeforces.ru \ Архив](#)
[298С. Игра на четность](#)

[codeforces.ru \ Архив](#)
[822В. Решение кроссворда](#)

Вася и стикер

Автор: Иван Кудласевич
[ej \ 25-08](#)

Шпиону Василию дали задание проникнуть в компьютерную систему врага. Под покровом ночи Василий получил доступ к одному из компьютеров вражеской системы. Для подключения к системе нужен пароль, которым, по разведанным, является сумма двух чисел. К счастью для Василия, он обнаружил стикер с двумя числами, любезно приклеенный к обратной стороне монитора.

Василий понял, что сам не справится, поэтому обращается к Вам за помощью. Он даёт Вам два числа, из которых нужно получить пароль.

Формат ввода: Две строки, в каждой по одному неотрицательному числу. Строки содержат не менее 1 и не более 255 символов. Возможны лидирующие нули.

Формат вывода: Искомая сумма.

input.txt

911111
111111

output.txt

1022222

26. Поиск слов в тексте. Лексикографический порядок. Процедура *continue*

Марсианские разведчики

Автор: Александр Кулицкий

ej \ 26-01

Однажды в 2106 году Землю решили атаковать марсиане. На Землю был послан корабль шпионов. Марсиане заранее договорились о том, что с момента выгрузки на планету они не должны разговаривать на марсианском языке.

Когда марсиане прилетели на Землю, командир приказал им (естественно, на земном языке) построиться в шеренгу по одному. Марсиане были ужасно неподготовлены к земному языку, они неправильно поняли своего командира, и построились все в один ряд, друг за другом.

Помогите марсианам построиться правильно, т.е. каждый марсианин должен стоять в отдельном ряду, также не должно быть пустых рядов.

Формат ввода: в единственной строке записаны имена разведчиков через один или несколько пробелов. Лишние пробелы могут быть в начале и в конце строки.

Формат вывода: в выходной файл запишите имена всех разведчиков в формате, приведённом ниже. Выходной файл не должен содержать пробелов.

input.txt

abc ab a abcd

output.txt

abc

ab

a

abcd

ej

2017-08-13 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ Е. Сергей и подстроки

Двоечник

Автор: Игорь Макеенко

ej \ 26-05

Вася плохо учится. Учительница задала ему на дом написать предложение и сказала, что если там будут ошибки, то она поставит ему 1. Вася не хочет получать 1. Он попросил Вас исправить его предложение.

Вам дано предложение Васи (длина не более 255 символов). Все слова в предложении разделены пробелами. Вы должны заменить все предлоги «by» на предлоги «on». Все предложения состоят только из маленьких латинских букв и точки в конце.

Формат ввода: исходная строка.

Формат вывода: искомая строка.

input.txt

by street.

output.txt

on street.

Первый робот

Автор: Владимир Миранович

ej \ 26-02

Первый робот, которого создал человек, умел читать только слова-палиндромы, а при чтении обычных слов робот выходил из строя.

Для того, чтобы робот читал и не выходил из строя, ученым требовалось выбирать из текста слова палиндромы. А так как у ученых не было времени этим заниматься, они попросили вас сделать это за них.

Формат ввода: одна строка, содержащая слова разделенные пробелами. Максимальная длина строки равна 253 символам.

Формат вывода: столбец из палиндромов. Если таких слов нет, тогда ничего не выводите.

Комментарий: в программе можно использовать не более одной строковой переменной.

input.txt

aba ab a abcd bb

output.txtaba
a
bb**Меньшее из частейших**

Автор: Владимир Миранович

ej \ 26-03

Прочитайте строку, состоящую из слов, разделённых пробелами. Максимальная длина строки равна 253 символам.

Выведите в файл слово, которое встречается в тексте наибольшее количество раз. Если таких слов несколько, выведите меньшее по алфавиту.

input.txt

aba ac aba de a de

output.txt

aba

dl.gsu.by \ Программирование – начинающие (Р/О)
21-30 \ 4 \ 4 - "Игра в слова"

ej

2017-09-03 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ С. Опасный бой

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2014 \ День 2 \ 1 - "Шифр Цезаря"

dl.gsu.by \ Олимпиады по информатике
Российская (ком.,Inet) \ 2008 \ Базовые \ Олимпиада_7 (12 января 2008) \ 6 -
"Лексикографически минимальное число"

codeforces.ru \ Тренировки
2012-2013 Всероссийская командная олимпиада школьников по программированию,
интернет-тур + отборы регионов (ВКОШП 12, Интернет-тур) \ Н. Ребус

Первоклассный хакер

Авторы: Сергей Коноплич, Вячеслав Лукьяненко
[ej \ 26-04](#)

Агенту Даниилу дали задание по взлому вражеской организации.

С помощью своего недюжинного обаяния агенту Даниилу удалось выяснить, что паролем от вражеской базы данных является максимальное число, которое можно составить из n чисел. Но агент Даниил понял, что сам не справится, поэтому он обращается к вам за помощью.

Вам даны n чисел, из которых нужно составить корректный пароль.

Формат ввода: в первой строке входного файла находится целое число n ($n \leq 250$). Далее - n строк, в каждой из которых находится одно число a_i , которое является частью пароля, $a_i \leq 10^{250}$.

Формат вывода: одно число – ответ на задачу. Не добавляйте перевод строки в конце числа.

genpass.in

3
348
352
1

passgen.out

3523481

codeforces.ru \ Архив
777D. Облако хэштегов

codeforces.ru \ Тренировки
2011-2012 Командный чемпионат школьников Санкт-Петербурга по программированию
(XIX) \ С. Номер страницы

27. Процедуры *insert*, *delete*

Корректор текста

Городская олимпиада по программированию, КЮП

Мозырь, 4 марта 2000 г.

[ej \ 27-01](#)

Дана строка, состоящая из букв и символов "#". Символ "#" означает команду: "стереть предыдущий символ".

Откорректируйте строку согласно описанному правилу, двигаясь в строке слева направо.

Комментарий: в программе можно использовать не более одного оператора цикла.

Формат ввода: исходная строка символов.

Формат вывода: откорректированная строка.

input.txt

AB###CDE#F

output.txt

CDE#F

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 5 - 8 кл \ 5 - "Редактор текста"](#)

[codeforces.ru \ Архив](#)

[1194C. Om S к T](#)

Письмо

Миниолимпиада КЮП

Мозырь, 8 декабря 1996 г.

Соавтор тестов: Иван Кудласевич

[ej \ 27-02](#)

Дан текст-письмо, состоящий из одного или нескольких предложений на английском языке с возможными лидирующими и концевыми пробелами.

Каждое предложение содержит слова, разделённые пробелами, начинается с большой буквы и заканчивается точкой. Других знаков препинания в предложениях нет.

В письме могут встречаться имена собственные вида «*Фамилия Имя Отчество*». Других слов с большой буквы кроме первого слова предложения и включений «*Фамилия Имя Отчество*» в предложениях нет.

Замените все включения «*Фамилия Имя Отчество*» на «*Фамилия Инициалы*».

Формат ввода: одна строка с исходным письмом.

Формат вывода: одна строка с откорректированным письмом.

Комментарий: данные примеров ввода и вывода располагаются в файлах в одну строку.

input.txt

Please convey my warm regards to
Ivanov Ivan Ivanovich.

output.txt

Please convey my warm regards
to Ivanov I.I.

28. Функция `pos`. Ограничение длины в строковой переменной. Типизированные константы. Процедуры `str`, `val`. Оператор `in`

А что в скобках

Автор: Максим Веремеев

[ej \ 28-01](#)

Из файла ввода прочитайте строку, содержащую произвольные символы и пару круглых скобок. Выведите в файл содержимое круглых скобок.

В программе нельзя использовать циклы.

input.txt

length (size)

output.txt

size

[dl.gsu.by \ Олимпиады по информатике](#)

[Гом. обл. младшие \ 2003 \ День 1 \ 3 - "Игра"](#)

Сломанный калькулятор

Авторы: Дмитрий Лещенко, Олег Любинский

[ej \ 28-02](#)

У одного известного программиста на калькуляторе сломалась кнопка умножения, когда ему нужно срочно было умножить два целых числа.

Помогите написать для него программу, которая из файла ввода читает строку, содержащую два числа и знак "*" между ними, а затем вычисляет произведение введенных чисел.

Формат ввода: строка с числами, разделёнными знаком "*".

Формат вывода: искомое произведение. Результат выведите в файл, разделяя пробелом группы из трёх разрядов.

Произведение не больше *maxlongint*.

input.txt

6400*1280

output.txt

8 192 000

Ленивый Вовочка

Авторы: Артём Бобр, Станислав Петров

[ej \ 28-03](#)

Вовочка, как всегда, не сделал домашнее задание по алгебре. Его учительница по этому предмету очень строгая, и требует от своих учеников своевременного выполнения домашнего задания. И, как назло, она вызвала Вовочку к доске, чтобы тот написал решение одного из номеров, заданных ею на дом. В этой задаче необходимо вычислить значение арифметического выражения. Задача была довольно проста, но Вовочка до такой степени растерялся, что попросил вас помочь ему с решением задачи.

Формат ввода: из файла ввода вам необходимо прочитать строку, которая состоит из чисел, разделённых ровно одним знаком '+', либо '-'. Длина строки не менее 3 и не более 250 символов.

Формат вывода: в файл вывода выведите результат вычислений.

Абсолютные значения промежуточных и конечного результатов не превышают 2 000 000 000.

input.txt

23+4-12+210+5-1-2

output.txt

227

Калькулятор

Городская олимпиада по программированию, КЮП

Мозырь, 13 марта 1999 г.

[ej \ 28-04](#)

Вычислите результат арифметических операций, произведённых на калькуляторе.

Формат ввода: первая строка – исходное число.

Строки с номерами 2.. $(n-1)$ содержат по два слова, разделённые пробелом: первое – название арифметической операции (*plus* – сложение, *minus* – вычитание, *divide* – целочисленное деление, *multiply* – умножение), второе – число.

Строка с номером n содержит слово *end*. $n \leq 11$.

При вводе используются целые числа из интервала 1..9, записанные словами: *one*, *two*, *three*, *four*, *five*, *six*, *seven*, *eight*, *nine*.

Формат вывода: число, записанное в десятичной системе счисления, - результат произведённых на калькуляторе операций.

input.txt

four
plus two
divide three
multiply five
end

output.txt

10

[codeforces.ru \ Архив](#)

[831В. Раскладки клавиатуры](#)

Длинные числа

Авторы: Георгий Коноплич, Егор Казаченко

[ej \ 28-05](#)

У Антона много братьев, а так как он самый старший, то братья часто просят Антона помочь им решать домашнее задание по математике. На этот раз один из братьев попросил у Антона сложить пару чисел, но числа оказались слишком длинные, и Антон решил написать программу. Помогите Антону написать программу по вычислению суммы чисел.

Формат ввода: две строки, в каждой по одному неотрицательному числу. Строки могут включать в себя пробелы и содержат не более 255 символов.

Формат вывода: искомая сумма. Результат выведите в файл, разделяя пробелом группы из трёх разрядов.

input.txt

```
1 234 560
12 300
```

output.txt

```
1 246 860
```

[dl.gsu.by \ Методы алгоритмизации](#)

[Олимпиады по информатике \ Реализация \ Двумерный массив \ 07_NEE - "Building for UN"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[GCSW \ 2001 \ Начинающие \ День 2 \ 5 - "Кладоискатели"](#)

Комментарий: полагайте, что выражение может содержать числа из нескольких разрядов

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 5 - 7 кл \ 9 - "Электронный журнал"](#)

[dl.gsu.by \ Методы алгоритмизации](#)

[Олимпиады по информатике \ Реализация \ Строки \ 07_CERC - "Reaux! Sham! Beaux!"](#)

[codeforces.ru \ Архив](#)

[791C. Мишка и различные имена](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[Гомельская обл. \ 2010 \ День 1 \ 3 - "Пробное задание"](#)

29. Двумерные массивы. Задачи с открытыми тестами

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. ком. младшие \ 2003 \ 2 - "Дом "*

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 9 \ 2 - "Отличники"*

*dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 2 \ 3 - "Максимальная высота местности"*

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 5 \ 2 - "Соревнование"*

*dl.gsu.by \ Программирование – начинающие (P/O)
GCSW \ 2002 \ Начинающие, ком. \ 3 - "Фото"*

Интересная матрица

Автор: Иван Киринский

[ej \ 29-01](#)

Прочитайте n , $n \leq 10$. Заполните массив a так, как показано в примере. Вывод данных должен проводиться так, как показано в примере фрагмента программы. В программе нельзя использовать оператор *if*.

Пример фрагмента программы:

```
...
for i:=1 to n do
begin
  for j:=1 to n-1 do write(a[i,j], ' ');
  writeln(a[i,n]);
end;
...
```

input.txt

5

output.txt

1	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	0

Песочные часы

Автор: Георгий Коноплич

ej \ 29-02

Прочитайте n , $n \leq 250$. Заполните массив a так, как показано в примере. Вывод данных должен проводиться так, как показано в примере фрагмента программы.

Пример фрагмента программы:

```
...
for i:=1 to n do
begin
  for j:=1 to n-1 do write(a[i,j], ' ');
  writeln(a[i,n]);
end;
...
```

input.txt

5

output.txt

```
1 1 1 1 1
0 1 1 1 0
0 0 1 0 0
0 1 1 1 0
1 1 1 1 1
```

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 5 - 8 кл \ 3 - "Квадраты"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 5 - 8 кл \ 3 - "Квадратное колесо - 2"

Построение

Автор: Алексей Галай

Соавтор тестов: Максим Клещёв

ej \ 29-03

Прочитайте n , $n \leq 50$. Выведите квадратную матрицу, заполненную согласно примеру вывода. Разделяйте числа одним пробелом.

input.txt

5

output.txt

```
3 3 3 3 3
3 2 2 2 3
3 2 1 2 3
3 2 2 2 3
3 3 3 3 3
```

Спираль

Авторы: Дмитрий Шарков

[ej \ 29-04](#)

Прочитайте n , $n < 10$ Выведите квадратную матрицу, заполненную согласно примеру вывода.

input.txt

5

output.txt

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

[codeforces.ru \ Архив](#)

404A. Валера и X

[codeforces.ru \ Архив](#)

389B. Лиса и крест

[codeforces.ru \ Тренировки](#)

2010-2011 командный чемпионат школьников Санкт-Петербурга по программированию (XVIII) \ С. Шоколад

[dl.gsu.by \ Олимпиады по информатике](#)

Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 5 - 7 кл \ 3 - "Количество единиц"

[dl.gsu.by \ Олимпиады по информатике](#)

Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 5 - 7 кл \ 5 - "Прямоугольник в прямоугольнике"

[dl.gsu.by \ Олимпиады по информатике](#)

Гомельская обл. \ 2010 \ День 1 \ 2 - "Анализ изображения"

[dl.gsu.by \ Олимпиады по информатике](#)

Гомельская обл. \ 2011 \ День 1 \ 1 - "Беговая дорожка"

Периметр площадки

Авторы: Евгений Костюкевич, Алексей Ропан, Пётр Майоров

[ej \ 29-05](#)

После разминки учитель по физкультуре в школе № 7 дал 11"А" классу задание. Они должны были пробежать вокруг площадки, представляющей собой на карте местности связную фигуру, и находящейся в матрице $N \times M$ ($3 \leq N, M \leq 100$), без пустоты внутри нее.

Самый умный ученик в классе - Вася - захотел правильно рассчитать свои силы, узнав длину беговой дорожки – периметра площадки. Но подсчитать их он никак не мог, и поэтому обратился за помощью к вам.

Формат ввода:

В первой строке даны числа N и M. В последующих N строках дается M чисел, разделённых одним пробелом: 0 – пустое пространство, и 2 - площадка.

Гарантируется, что можно обежать вокруг площадки.

В случаях, когда ученику придётся бежать по одному месту, учитывать расстояние нужно только один раз. Например:

```
5 6
0 0 0 0 0 0
0 2 2 2 2 0
0 2 0 0 0 0
0 2 2 2 2 0
0 0 0 0 0 0
```

В этом случае клетки 3,3; 3,4; 3,5; нужно учитывать только один раз.

Формат вывода:

В первой и единственной строке выведите одно число - расстояние, которое нужно пробежать.

input.txt

```
4 3
0 0 0
0 2 0
0 0 0
0 0 0
```

output.txt

```
8
```

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2001 \ Начинающие \ День 2 \ 3 - "Сканер"

Комментарий: полагайте, что ячейка граничит с 8 соседними ячейками, нулевые клетки могут граничить между собой

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 9 - 11 кл \ 2 - "Транспонирование"

dl.gsu.by \ Методы алгоритмизации

Олимпиады по информатике \ Реализация \ Двумерный массив \ 08_Gust - "Поиск прямоугольников"

Мебель

Городская олимпиада по программированию, КЮП

Мозырь, 13 марта 1999 г.

[ej \ 29-06](#)

План помещения представлен в виде прямоугольника размером $m \times n$ клеток, $m \leq 300$, $n \leq 300$.

В помещении находятся один или несколько предметов мебели, изображённых на плане в виде прямоугольников, закрашенных единицами. Предметы не соприкасаются друг с другом по вертикали, горизонтали, диагонали.

Определите количество предметов, расположенных в помещении.

Формат ввода: первая строка содержит два числа, разделённые пробелом. Строки с номерами 2..($m+1$) содержат план помещения.

В каждой строке находится n чисел (0 или 1), разделённых пробелами. 0 означает, что клетка не занята предметом, 1- занята.

Формат вывода: число – количество предметов в помещении.

input.txt

```
5 4
1 0 0 0
1 0 1 0
0 0 0 0
1 1 1 0
1 1 1 0
```

output.txt

```
3
```

[ej](#)

2017-09-03 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ F. Просвещение

Световая реклама

Автор: Александр Кулицкий

[ej \ 29-07](#)

Щит световой рекламы состоит из лампочек, расположенных в m рядов по n штук. Некоторые лампочки перегорели.

Задано изображение, состоящее из h рядов по w точек, которое нужно отобразить на этом щите.

Требуется выяснить, какое минимальное количество перегоревших лампочек k нужно заменить, чтобы можно было отобразить на щите световой рекламы заданное изображение.

Формат ввода:

```
n m
a1,1 a1,2 ... a1,n
...
am,1 am,2 ... am,n
w h
```


$$b_{1,1} \ b_{1,2} \ \dots \ b_{1,w}$$

$$\dots$$

$$b_{h,1} \ b_{h,2} \ \dots \ b_{h,w}$$

Здесь m , n , h , w – размеры двух матриц; a_{ij} и b_{ij} – элементы матриц, задающих элементы щита и заданного изображения.

Если $a_{ij}=1$, то это исправная лампочка, если $a_{ij}=0$, то это перегоревшая лампочка.

Если $b_{ij}=1$, то это светящийся элемент изображения, если $b_{ij}=0$, то это не светящийся элемент изображения.

Числа в строке разделяются одним пробелом.

Формат вывода:

$$k$$

Здесь k – минимальное количество лампочек, которые нужно заменить.

Ограничения:

$$0 < n, m \leq 100$$

$$0 < w \leq n$$

$$0 < h \leq m$$

input.txt

```
10 7
1 1 1 0 1 1 1 1 1 1
0 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
0 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 1 1
9 5
0 1 1 0 0 1 0 0 1
1 0 0 1 0 1 0 1 0
1 0 0 1 0 1 1 0 0
1 0 0 1 0 1 0 1 0
0 1 1 0 0 1 0 0 1
```

output.txt

```
0
```

[codeforces.ru \ Архив](https://codeforces.ru/problemset/problem/710/C)

710C. Магический нечётный квадрат

[codeforces.ru \ Архив](https://codeforces.ru/problemset/problem/431/B)

431B. Очередь в душ

[dl.gsu.by \ Олимпиады по информатике](https://dl.gsu.by/olympiads/informatika)

Гомельская обл. \ 2016 \ День 1 \ 1 - "Морской бой"

*dl.gsu.by \ Олимпиады по информатике
Balkan Junior OI \ 2007 \ День 1 \ 2 - "Piles of books"*

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2009 \ October \ Gold \ 4 - "Papaya Jungle"*

*dl.gsu.by \ Методы алгоритмизации
Олимпиады по информатике \ Реализация \ Двумерный массив \ 08_Gust - "Квадрат"*

*ej
2017-06-18 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра
Кривошеева \ F. Сергей и морской бой*

*codeforces.ru \ Архив
828B. Чёрный квадрат*

*codeforces.ru \ Архив
837B. Флаг Берляндии*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2013 \ День 2 \ 4 - "Лабиринт (с открытыми тестами)"*

30. Система координат на плоскости. Длина отрезка. Координаты середины отрезка. Точность вычислений. Прямоугольник. Точка внутри прямоугольника со сторонами, параллельными осям координат. Функции *trunc*, *int*, *frac*. Остаток от деления вещественных чисел. Вычисление степени с вещественным показателем

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 10 \ 3 - "Путешественники"*

*dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 4 \ 3 - "Города"*

*dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 3 \ 3 - "За грибами"*

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 9 \ 3 - "Кто быстрее?"

dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 1 \ 3 - "Три друга "

dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 1 \ 3 - "Скрудж Мак Дак"

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 4 \ 3 - "Тест"

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 6 \ 3 - "Поход"

dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 2 \ 2 - "Поезд"

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 7 \ 4 - "Путь разносчика газет"

dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 7 \ 3 - "Турпоход"

ej
2017-06-11 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Игоря
Макееенко и Андрея Мищенко \ С. Костя и дерево

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 1 \ 4 - "Футбольное поле"

Комментарий: стороны поля параллельны осям координат

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 1 - 4 кл \ 17 - "Задача 17"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 3 - "Вывести расстояния"

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2002 \ День 1 \ 4 - "Города"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 1 \ 3 - "Деревня Васюльки"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2005 \ Городская (3 декабря) \ Младшие \ 3 - "Дятлик спасатель"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 5 - 7 кл \ 4 - "Неутомимый Карлсон - 2"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. ком. младшие \ 2003 \ 4 - "Место встречи"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 5 - 8 кл \ 4 - "Длина дистанции"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 5 - 8 кл \ 4 - "Количество мостов"*

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2009 \ December \ Bronze \ 2 - "Lonesome Partners"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2002 \ День 2 \ 3 - "Забор"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2003 \ День 1 \ 4 - "Магазин"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 3 - "Точки"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2002 \ Младшая группа \ 4 - "Эпидемия"*

[codeforces.ru \ Архив](https://codeforces.ru/archive/problem/424B)
424B. Город-миллионер

[codeforces.ru \ Архив](https://codeforces.ru/archive/problem/404B)
404B. Марафон

31. Неравенства треугольника. Виды треугольников. Площадь треугольника. Метод трапеций

informatics.mccme.ru
Задача №2936. Гипотенуза

Равнобедренный треугольник

Автор: Владислав Собин

[ej \ 32-01](#)

Определите, является ли треугольник равнобедренным.

Формат ввода: три строки содержат координаты вершин треугольника.

Формат вывода: "yes" или "no".

input.txt

```
-1.5 0  
1.5 0  
0 2
```

output.txt

```
yes
```

Прямоугольный треугольник

Автор: Владислав Собин

[ej \ 32-02](#)

Определите, является ли треугольник прямоугольным.

Формат ввода: три строки содержат координаты вершин треугольника.

Формат вывода: "yes" или "no".

input.txt

```
0.5 0  
4.5 0  
0.5 3
```

output.txt

```
yes
```

Властелины треугольников

Автор: Андрей Левковский

[ej \ 32-03](#)

Однажды Петя и Вася, насмотревшись фильма "Властелин колец", решили вырезать из бумаги круги, которые будут похожи на те самые кольца. Но так как они не умеют вырезать, то у них получаются не круги, а треугольники.

И тогда решили Петя и Вася с горя проверить, являются ли треугольники, которые они вырезали, подобными. Помогите Пете и Васе.

Формат ввода: первые три строки содержат координаты вершин первого треугольника, следующие три строки - координаты вершин второго треугольника.

Формат вывода: коэффициент подобия с точностью два знака после запятой, если треугольники подобны, или "no".

input.txt

```
0 0
1.5 0
0 2
0 0
0 4
3 0
```

output.txt

```
2.00
```

[codeforces.ru \ Группы \ КЮП](#)

[usaco \ А. Строительство забора](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[GCSW \ 1999 \ Начинающие \ День 1 \ F - "Площадь заповедника"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[GCSW \ 1999 \ Начинающие, ком. \ F - "Раздел царства"](#)

Комментарий: результат должен быть округлён до ближайшего целого

32. Круг и окружность. Длина окружности. Площадь круга

[dl.gsu.by \ Олимпиады по информатике](#)

[GCSW \ 2002 \ Начинающие \ День 2 \ 4 - "Соревнование на меткость"](#)

[dl.gsu.by \ Олимпиады по информатике](#)

[GCSW \ 2002 \ Начинающие \ День 1 \ 2 - "Лесник"](#)

Оружие

Автор: Александр Кулицкий

[ej \ 32-04](#)

Учеными был изобретён новый вид оружия. Действовал он так: на землю сбрасывали два снаряда в форме круга. Оружие срабатывало, если снаряды после падения пересекались хотя бы в одной точке.

По заданным координатам падения центров каждого снаряда определите, сработает ли оружие.

Формат ввода: координаты центров падения снарядов, и их радиусы.

Формат вывода: "yes", если оружие сработает, и "no" в обратном случае.

gun.in

1 2 3
5 3 3

gun.out

yes

[dl.gsu.by \ Олимпиады по информатике](#)

[GCSW \ 2003 \ Начинающие \ День 2 \ 3 - "Веревочка](#)

[dl.gsu.by \ Программирование – начинающие \(P/O\)](#)

[1-10 \ 2 \ 3 - "Парашютист"](#)

[dl.gsu.by \ Программирование – начинающие \(P/O\)](#)

[11-20 \ 10 \ 3 - "Задача о мишени"](#)

Пересечение

Миниолимпиада КЮП

Мозырь, 8 декабря 1996 г.

[ej \ 32-05](#)

Заданы две окружности своими центрами и радиусами: $O_1(x_1, y_1)$, r_1 и $O_2(x_2, y_2)$, r_2 . Определите, пересекаются ли окружности. Окружности пересекаются, если имеют хотя бы одну общую точку.

Формат ввода: две строки с тройками чисел x y r - координатами центров и радиусами.

Формат вывода: "yes" или "no".

input.txt

1 2 3
5 3 3

output.txt

yes

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Начинающие \ День 1 \ 3 - "Геометрическая"

Том и Джерри 2

Автор: Алексей Ропан

ej \ 32-06

Хозяйка в последнее время стала замечать слишком много мышиных норок. Из-за этого она захотела выбросить ленивого Тома на улицу.

Кот, недолго думая, пошёл к Джерри и попросил сократить число своих норок. Джерри понял, что если Тома выгонят, то у хозяйки появится другой кот, который, скорее всего, будет быстрее и хитрее Тома, и поэтому согласился на просьбу кота.

Лунки отличаются друг от друга радиусами, и Джерри хочет избавиться от лунок с меньшей площадью входа. Джерри снова нужна ваша помощь.

Формат ввода:

n количество норок ($1 \leq n \leq 1000$)

a_1

a_2 a_i - радиус i -ой норки ($1 \leq a_i \leq 30000$)

...

a_n

m количество норок, которое нужно сократить ($0 \leq m \leq n$)

Формат вывода: Выведите суммарную площадь норок от которых надо избавиться, с точностью два знака после запятой.

TandJ2.in

5
2
3
5
4
7
3

TandJ2.out

91.11

dl.gsu.by \ Олимпиады по информатике
GCSW \ 1999 \ Начинающие \ День 2 \ F - "Метание дротиков"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 1999 \ Начинающие, ком. \ D - "Военный полигон"

Том и Джерри

Автор: Алексей Ропан

[ej \ 32-07](#)

Всем, наверное, известны приключения кота Тома и мышонка Джерри. И вот, в очередной раз Том загнал Джерри в круглую комнату.

Джерри был уверен, что если побежать вдоль правой стены, то можно добежать до норки за t секунд, и это его вполне устраивало.

В то же время Том был уверен, что если побежать за мышонком, то его не догнать (Том уже много раз пытался) и он также был убежден, что если бежать через комнату, то можно попасть под веник хозяйки. Поэтому Том решил бежать вдоль левой стены.

Джерри, заметив стратегию Тома, решил не останавливаться, так как был сильно испуган. Но его всё же тревожило - успеет ли он добежать (успеет он в том случае, если окажется около норки раньше Тома). Помогите Джерри.

Формат ввода: $r \ t \ v_t \ v_j$

r радиус комнаты (в метрах)
 t время, за которое Джерри добежит до норки (в секундах)
 v_t скорость Тома (метров в секунду)
 v_j скорость Джерри (метров в секунду)

Комментарий: все числа положительные и не превышают 10000.

Формат вывода: Если Джерри успеет, следует вывести "Yes, T1", где T1 - время, за которое Том добежит до норки, иначе - "No, T2", где T2 - время, через которое они встретятся (время выводить с точностью три знака после запятой).

TandJ.in

2 1 3 4

TandJ.out

Yes, 2.855

[codeforces.ru \ Тренировки](#)

[2010-2011 командный чемпионат школьников Санкт-Петербурга по программированию \(XVIII\) \ Д. Луна](#)

[codeforces.ru \ Тренировки](#)

[2012-2013 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов \(ВКОШП 12, Интернет-тур\) \ Я. День рождения викинга](#)

33. Объявление процедур и функций. Процедура exit

[codeforces.ru \ Тренировки](#)

[2013-2014 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов \(ВКОШП 13, Интернет-тур\) \ В. Ставка](#)

codeforces.ru \ Тренировки

2011-2012 Командный чемпионат школьников Санкт-Петербурга по программированию (XIX) \ А. Летопись

Ферзь

Городская олимпиада по программированию, КЮП

Мозырь, 13 марта 1999 г.

[ej \ 33-01](#)

На шахматной доске находится белый ферзь и несколько других фигур. Определите, какие из фигур чёрных находятся под боем.

Формат ввода: первая строка – координаты ферзя, вторая строка – координаты одной или нескольких белых фигур, третья строка – координаты одной или нескольких чёрных фигур.

Координаты в строках разделяются между собой пробелом. Латинские символы в координатах принадлежат нижнему регистру.

Максимальное количество фигур каждого цвета на доске – 16.

Формат вывода: строка, содержащая координаты чёрных фигур, находящихся под боем, или сообщение "none", если таких фигур нет. Координаты разделяются между собой пробелом, отсортированы по возрастанию.

input.txt

```
c3
b2 h6 b6 e5 h1
g3 d7 b4 g7
```

output.txt

```
b4 g3
```

dl.gsu.by \ Олимпиады по информатике

USACO \ 2014 \ February \ Bronze \ 1 - "MIRROR"

Отгадывание кроссворда

АСМ Финал

1994 год

[ej \ 33-02](#)

Кроссворд состоит из прямоугольной сетки чёрно-белых квадратов и двух списков определений (или описаний). Один список описаний для слов, которые будут написаны слева направо по белым квадратам в строках, и другой список для слов, которые будут написаны вниз по белым квадратам в столбцах. Словом считается последовательность букв. Чтобы решить кроссворд, на белых квадратах сетки записываются соответствующие описаниям слова.

Описания соответствуют прямоугольной сетке по значениям последовательных чисел на "подходящих" белых квадратах. Белый квадрат с соседствующим слева или сверху чёрным квадратом является "подходящим". Белый квадрат без соседа слева или сверху также "подходящий". Никакие другие квадраты не нумеруются. Все квадраты на первой строке нумеруются. Нумерация начинается с 1 и продолжается последовательно по всем белым квадратам первой строки, затем по подходящим белым квадратам второй строки, затем по

подходящим белым квадратам третьей строки и так далее по всем оставшимся строкам кроссворда.

Нужно написать программу, которая берёт несколько решённых кроссвордов как входные данные и выводит списки слов по горизонтали и по вертикали, которые составляют решение кроссворда.

Формат ввода: каждый кроссворд на входе начинается строкой, содержащей два целых числа r и c ($1 \leq r \leq 10$ и $1 \leq c \leq 10$), где r (первое число) - количество строк в кроссворде и c (второе число) - количество столбцов. r строк ввода, которые следуют далее, содержат c символов (исключая перевод строки), которые составляют кроссворд. Каждый из символов либо буква, которая является частью слова, либо символ "*", который указывает на чёрный квадрат. Конец ввода обозначен строкой, состоящей из единственного числа 0. Количество кроссвордов во вводе не более 1000.

Формат вывода: вывод для каждого кроссворда состоит из его номера (puzzle #1, puzzle #2, и т.д.) и списка слов. Слова в каждой строке должны быть по одному в порядке их встречи в кроссворде. В случае, если списки слов пустые, все равно следует выводить Across и Down.

Кроссворды в выводе разделяются друг от друга двумя пустыми строками.

resolve.in

```
2 2
AT
*O
6 7
AIM*DEN
*ME*ONE
UPON*TO
SO*ERIN
*SA*OR*
IES*DEA
0
```

resolve.out

```
puzzle #1:
Across
  1.AT
  3.O
Down
  1.A
  2.TO

puzzle #2:
Across
  1.AIM
  4.DEN
  7.ME
  8.ONE
  9.UPON
 11.TO
 12.SO
 13.ERIN
 15.SA
 17.OR
 18.IES
 19.DEA
Down
  1.A
  2.IMPOSE
  3.MEO
  4.DO
  5.ENTIRE
```

6.NEON
9.US
10.NE
14.ROD
16.AS
18.I
20.A

Тест светодиодов

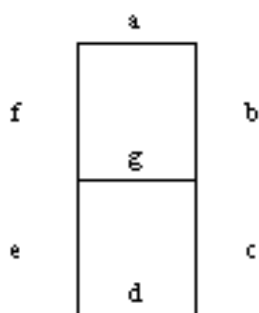
1995 East-Central Regionals ACM International Collegiate Programming Contest

Sponsored by Microsoft

[ej \ 33-03](#)

Много электронных устройств имеют цифровые индикаторы, которые используют наборы светодиодов. Элемент обычного цифрового дисплея использует семь прямоугольных светодиодов с сегментами, расположенными следующим образом:

Seven Segments Configuration



Цифры отображаются с помощью включения различных комбинаций из семи сегментов в соответствии со следующей таблицей:

Свечение сегментов в цифрах (Y - да, N - нет)

	a	b	c	d	e	f	g
0	Y	Y	Y	Y	Y	Y	N
1	N	Y	Y	N	N	N	N
2	Y	Y	N	Y	Y	N	Y
3	Y	Y	Y	Y	N	N	Y
4	N	Y	Y	N	N	Y	Y
5	Y	N	Y	Y	N	Y	Y
6	Y	N	Y	Y	Y	Y	Y
7	Y	Y	Y	N	N	N	N
8	Y	Y	Y	Y	Y	Y	Y
9	Y	Y	Y	Y	N	Y	Y

Например, цифра три будет отображена как совокупность сегментов {a,b,c,d,g}.

Задача ниже абстрагирована из подобной задачи, встреченной в отделе проверки качества входящих частей на фабрике по производству микроволновых печей. Приходящие

семисегментные светодиодные индикаторы проверяются на их корректное функционирование с помощью автоматического оборудования, работающего под программным управлением.

Замечание: решение задачи ниже ни достаточно, ни необходимо для тестирования в реальных условиях. Вы должны создать программу, которая будет наблюдать за последовательными тестами состояния свечения семи сегментов одноцифрового дисплея и будет решать, является ли последовательность свечений верной последовательностью "обратного отсчёта".

К сожалению, в индикаторах, которые исследует ваша программа, некоторые из элементов могут выйти из строя в начале теста, и некоторые из элементов могут выйти из строя во время теста. Ни один из сегментов не сгорит во включенном состоянии, и ни один из вышедших из строя сегментов не будет восстановлен. Тем не менее, вы должны "прочитать" сквозь двусмысленность, присутствующую в повреждённых дисплеях, просматривая на дисплее обратный отсчёт через одно или несколько значений.

Формат ввода: ввод состоит из серий наборов данных, имеющих следующий формат. Первая строка - натуральное одиночное число больше 0 и меньше 11. Число упорядочено по левому краю строки ввода. Следующие N строк - в каждой строке есть последовательность из семи состояний. Символы Y/N показывают состояние свечения сегментов индикатора. Последовательные строки предположительно представляют последовательность обратного отсчёта. Первый из семи символов - это свечение сегмента "а". Конец данных отмечается нулевым набором данных с нулём на первой строке. Нет информации о том, где последовательность обратного отсчёта начинается для данного набора данных. Нет информации, которая переносится с одного набора в другой, каждый набор данных представляет отдельный светодиодный индикатор.

Формат вывода: для каждой предположительной последовательности обратного отсчёта во входном файле вы должны ответить MATCH или MISMATCH в зависимости от того, может ли эта последовательность быть верной последовательностью обратного отсчёта где-нибудь внутри последовательности {9, 8, 7, 6, 5, 4, 3, 2, 1, 0}, если неизвестное количество неисправных элементов присутствует в начале теста и, возможно, сгорит во время тестирования.

ledtest.in

```
1
YYYYYYY
2
NNNNNNN
NNNNNNN
2
YYYYYYY
YYYYYYY
3
YNYYYYY
YNYNYY
NYNNYY
3
YNYYYYN
YNYYNYY
NYNNYYN
3
```

ledtest.out

```
MATCH
MATCH
MISMATCH
MATCH
MATCH
MISMATCH
MATCH
MATCH
```

```

YNYYYYN
YNYYNYN
NYYNYYN
4
YYYYYYY
NYYNNNN
NNYYYYN
NNNYNNN
3
NNNNNNN
YNNNNNN
NNNNYNN
0

```

Куб

Отборочные сборы к республиканской олимпиаде по программированию
среди учащихся внешкольных учреждений

Гомель

[ej \ 33-04](#)

Дан куб, каждая сторона которого раскрашена в различный цвет. Известна серия ходов его прохождения по прямоугольнику размером $k \times p$, $k \leq 10$, $p \leq 10$.

Найдите количество клеток одного цвета, которых больше всего отпечаталось на прямоугольнике после прохождения куба.

Формат ввода: Первая строка: k p . Вторая строка: стороны кубика. Могут быть окрашены в красный (R), жёлтый (Y), зелёный (G), синий (B), оранжевый (O) и голубой (C) цвета.

Окраска кубика задаётся строкой из 6 букв, разделённых пробелами, в следующей последовательности: низ, верх, север, юг, запад, восток.

Третья строка: последовательность ходов - строка со сторонами света в порядке перемещения кубика по прямоугольнику: север (N), юг (S), запад (W), восток (E).

Кубик начинает движение из левой верхней клетки поля. Число ходов ≤ 200 .

Формат вывода: искомое количество.

input.txt

```

3 2
R Y G B O C
E S W S E N N

```

output.txt

```

3

```

[codeforces.ru \ Архив](#)
[387C. Георгий и число](#)

[codeforces.ru \ Тренировки](#)
[2013-2014 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов \(ВКОШП 13, Интернет-тур\) \ D. Болезнь](#)

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2011 \ День 2 \ 3 - "Калькулятор"*

*informatics.mccme.ru
Задача №112093. Маленькая сказка о фиолетовом бобре*

34. Кратчайшее расстояние в таблице. Манхэттенское расстояние

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 7 \ 5 - "Бомбардировка"*

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 9 \ 5 - "Шахматный король"*

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 8 \ 5 - "Орел и Комар"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2003 \ День 2 \ 6 - "Game"*

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2016 \ День 2 \ 2 - "Магические свитки"*

35. Стек и дек

ИИ

Авторы: Александр Кулицкий, Алексей Ропан
ej \ 35-01

В недалеком будущем белорусские программисты изобрели искусственный интеллект (ИИ). Он мог ответить на любой вопрос, но отвечал он либо "Yes", либо "No", притом в качестве ответа он выдавал последовательность из квадратных, круглых, угловых и фигурных скобок.

В случае положительного ответа ИИ выдавал правильную скобочную последовательность, в противном случае неправильную. Вам поручили написать программу, которая будет расшифровывать ответ ИИ.

s - ответ III

Формат вывода: если ответ положительный, выведите "Yes", в противном случае - "No".

$$(\langle () \rangle [\]) [\{ \langle \rangle \}]$$

Yes

Авторы: Александр Кулицкий, Алексей Ропан

```

3
( ([ ] ) ( ) )
( ([ ] [ ] [ ] [ ] ( [ ] ) ( [ ] ) ( [ ] ) ( [ ] ) ) ( ) )
( ) ( ( ( ( ( ( ( [ ] [ ] [ ] [ ] [ ] [ ] ) ) ) ) ) ) )

```

Winner: 2

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2013 \ Day2 \ 2 - "Эволюция"*

*codeforces.ru \ Архив
1175B. Поймай переполнение!*

*dl.gsu.by \ Олимпиады по информатике
COCI \ 2012 \ coci_2011-12 \ 12_COCI_4 \ 3 - "KEKS"*

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2009 \ March \ Silver \ 3 - "Look Up"*

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2006 \ November \ Silver \ 1 - "Bad Hair Day"*

36. Очередь. Контейнер *pair*

Пятна

ej \ 30-01

На прямоугольной карте $m \times n$, $m, n \leq 10$, задано расположение нескольких пятен, закрашенных числом 1.

Пятном считается группа клеток, граничащих друг с другом по горизонтали или вертикали.

Определите количество пятен на карте.

Формат ввода: первая строка: m n – размеры таблицы. Остальные строки: таблица с форматом согласно примеру ввода.

Формат вывода: искомое количество.

input.txt

```
7 8
0 1 1 0 0 0 0 0
0 1 0 0 1 0 1 0
0 1 0 0 0 0 1 0
0 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0
0 1 1 0 1 0 0 0
0 1 1 0 1 1 1 0
```

output.txt

```
4
```

dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 1 \ 5 - "Африканская страна"

dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 1 \ 5 - "Азовское море"

dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 2 \ 5 - "Фермер"

dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 10 \ 5 - "Группы"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 1 \ 5 - "Поле (с подсказками)"

dl.gsu.by \ Олимпиады по информатике
USACO \ 2008 \ December \ Bronze \ 2 - "Bad Grass"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Начинающие \ День 2 \ 5 - "Минное поле (с подсказками)"

Лабиринт 2

Автор: Сергей Кулик
ej \ 36-01

В этой задаче вам надо всего лишь узнать: можно ли из одной точки некоторого лабиринта добраться до другой.

Формат ввода: в первой строке входного файла находится число n ($0 < n < 501$). Далее следует матрица $n \times n$ чисел, каждое из которых будет равно единице (клетка непроходима), или нулю (клетка свободна).

Формат вывода: выведите ":", если из клетки (1,1) можно пройти в клетку (n,n) или "(", если нельзя.

input.txt

```
5
0 1 1 1 1
0 0 0 0 0
1 1 0 1 1
1 1 0 1 1
0 0 0 0 0
```

output.txt

```
:)
```

Грядки

Четвертьфинальные соревнования

командного чемпионата мира по программированию ACM 98/99

Рыбинск, 6-7 октября 1998 г.

[ej \ 36-02](#)

Садовый участок, имеющий прямоугольную форму и разбитый на квадратные клетки со стороной 1 метр, имеет ширину n и длину m метров. На этом участке вскопаны грядки.

Грядкой называется совокупность клеток, удовлетворяющих условиям:

- из любой клетки этой грядки можно попасть в любую другую клетку этой же грядки, последовательно переходя по грядке из клетки в клетку через их общую сторону;
- никакие две грядки не пересекаются и не касаются друг друга ни по вертикальной, ни по горизонтальной сторонам клеток (касание грядок по углам клеток допускается).

Подсчитайте количество грядок на садовом участке.

Формат ввода: В первой строке исходного файла заданы два числа: n и m , разделённые одним или несколькими пробелами. Далее следуют n строк по m символов.

Символ "#" обозначает территорию грядки. Символ "." соответствует незанятой территории. Других символов в исходном файле нет. Размеры участка удовлетворяют условиям $1 \leq n, m \leq 200$.

Формат вывода: Выведите в выходной файл количество грядок на садовом участке.

garden.in

```
5 10
##.....#.
.#..#...#.
.###.....#.
..##.....#.
.....#.
```

garden.out

```
3
```

Китайские поля

Городская олимпиада по информатике

Мозырь, ноябрь 1998 г.

[ej \ 36-03](#)

Где только не живут люди! Те, кто немного знаком с географией, знают, что в Китайской Народной Республике большие площади земли представляют собой холмистую, сильно пересечённую местность. Иногда трудно представить, что на такой земле можно вырастить урожай, ведь дожди смывают со склонов холмов верхний плодородный слой. Однако коренные жители Китая уже очень давно нашли способ сделать эти земли плодородными и пригодными для выращивания урожая. На склонах холмов они устраивают террасы, то есть участки с горизонтальной поверхностью, на которых дождевая вода задерживается и не смывает плодородный слой.

Разница высот между соседними террасами может составлять несколько метров. Карта некоторой местности может быть задана в виде массива целых чисел, в котором каждое число определяет высоту некоторого квадратного участка местности.

Легко видеть, что если два квадрата имеют одинаковую высоту и граничат в одном из четырёх направлений – север, юг, запад, восток, - то они принадлежат одной и той же террасе.

Напишите программу, которая получает карту некоторой местности и подсчитывает количество различных террас, расположенных на ней.

Формат ввода:

```
k n
v[1,1] ... v[1,n]
v[2,1] ... v[2,n]
...
v[k,1] ... v[k,n]
```

Здесь k, n – количество строк и столбцов в карте ($1 \leq k, n \leq 100$); $v[i, j]$ – высота квадрата с координатами i, j .

Формат вывода: количество различных террас на местности.

Комментарий: высоты квадратов – целые положительные числа диапазона *longint*.

input.txt

```
5 5
1 1 1 1 1
1 3 3 1 1
1 3 5 3 1
1 1 3 3 1
1 1 1 1 1
```

output.txt

```
4
```

Все выходы

[ej \ 36-04](#)

Дана карта, в которой клетка равна 0, если она проходима, или 1, если непроходима.

Путник может перемещаться из одной проходимой клетки в другую, если они имеют общую сторону. Путник выходит из лабиринта, когда попадает в граничную клетку.

Формат ввода:

```
m n
a[1,1] a[1,2] ... a[1,n]
a[2,1] a[2,2] ... a[2,n]
...
a[m,1] a[m,2] ... a[m,n]
k l
```

m, n – количество строк и столбцов в карте лабиринта ($1 \leq m, n \leq 40$); $a[i, j]$ – информация о лабиринте;

k, l – исходное положение путника в проходимой клетке.

Формат вывода: слово "no" или список координат возможных выходов для путника, упорядоченный по возрастанию строк и столбцов.

input.txt

```
9 10
```

output.txt

```
1 6
```

```

1 1 1 1 1 0 1 1 1 1
1 0 0 0 0 0 1 1 1 1
1 0 1 1 1 1 1 1 0 0
1 0 0 0 0 0 1 1 0 1
1 1 1 0 1 1 0 0 0 1
1 1 1 0 1 1 0 1 1 1
1 1 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 0 1
1 1 1 1 0 1 1 1 1 1
7 6

```

```

3 10

```

Путешествие с конём

The 10th Latvian Olympiad in Informatics

Regional competition Problems

ej \ 36-05

Размеры прямоугольной размеченной квадратами доски $n \times m$. В нижнем левом квадрате доски (1,1) находится шахматный конь.

Конь может ходить только согласно шахматным правилам - движение может быть двумя квадратами горизонтально и затем одним вертикально, или двумя квадратами вертикально и одним горизонтально. Например, если $n=4$ и $m=3$, и конь находится в квадрате (2,1), то следующим может быть ход (1,3) или (3,3) или (4,2).

Для заданных положительных целых значений n, m, i, j ($n, m \leq 100, i \leq n, j \leq m$) вы должны определить вывод минимальное необходимое количество ходов коня для перемещения из начальной позиции (1,1) в квадрат (i,j). Если это невозможно, вы должны вывести "NEVAR".

Формат ввода:

$n \ m \ i \ j$

Формат вывода: количество ходов или сообщение об отсутствии пути.

Пример 1

knight.in

100 2 2 2

knight.out

NEVAR

Пример 2

knight.in

5 3 1 2

knight.out

3

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2002 \ Начинающие \ День 1 \ 5 - "Паук"

dl.gsu.by \ Олимпиады по информатике

Гом. обл. младшие \ 2003 \ День 1 \ 5 - "Индиана Джонс"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2005 \ Городская (3 декабря) \ Младшие \ 5 - "Дятлик в супермаркете"

Комментарий: допускается расположение живодёра в отделе игрушек

dl.gsu.by \ Олимпиады по информатике
USACO \ 2005 \ January \ BRONZE \ 4 - "SATEL"

dl.gsu.by \ Олимпиады по информатике
USACO \ 2007 \ April \ Silver \ 2 - "Catch That Cow"

codeforces.ru \ Архив
217A. Катание на Коньках

Скалолаз

Областная олимпиада школьников по информатике. День 1
 Гомель, 1999 г.
ej \ 36-06

Скалолаз собирается пройти некоторый гористый участок местности. Если на его пути появляется отвесная стена (спуск или подъём на скалу), скалолаз забивает страховочные скобы, продевает в них канат, и с помощью этих приспособлений преодолевает препятствие.

Скобы забиваются через равные промежутки, поэтому, чем выше препятствие (спуск или подъём), тем больше скоб расходуется. После преодоления препятствия скалолаз снимает канат и забирает его с собой, а скобы остаются забитыми в скалы. Чтобы подняться (или опуститься) на n метров, нужно забить n скоб.

Карта местности задаётся в виде массива целых чисел, в котором каждое число задаёт высоту в метрах некоторого квадратного участка местности. Напишите программу, которая определит, какое наименьшее количество скоб потребуется скалолазу, чтобы пройти из верхнего левого угла карты в нижний правый.

Скалолаз может перемещаться в одном из четырёх направлений - север, юг, запад, восток (не считая подъёмов и спусков).

Формат ввода:

n m
 $v[1,1] \dots v[1,m]$
 $v[2,1] \dots v[2,m]$
 ...
 $v[n,1] \dots v[n,m]$

Здесь n - размер карты по вертикали, m - размер карты по горизонтали ($1 \leq n, m \leq 20$), $v[i,j]$ - высота квадрата местности с координатами i,j ($1 \leq v[i,j] \leq 100$).

Формат вывода: k - минимальное количество скоб, которое потребуется скалолазу.

rockman.in

```
5 5
1 1 1 2 3
1 1 2 3 2
```

rockman.out

```
2
```

1	1	3	2	1
1	2	1	1	1
2	1	1	1	1

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2001 \ Младшая группа \ 6 - "Интересная игра"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 6 - "Айсберги"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие, ком. \ 8 - "Не пройти, не проехать..."*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-11 кл, 9 ноября \ 9 - 11 кл \ 6 - "Леса и горы"*

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Начинающие \ День 1 \ 6 - "Игра"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 6 - "Обобщённый конь"*

*dl.gsu.by \ Олимпиады по информатике
Канадская \ 2012 \ Day_2 \ 1 - "EDITOR"*

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2002 \ День 1 \ 5 - "Настольная Игра"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2014 \ День 1 \ 3 - "Выверенные прыжки"*

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2013 \ April \ Silver \ 1 - "GRAVITY"*

37. Кольцевая очередь. Восстановление пути. Контейнер vector

Лабиринт

Олимпиада по программированию

Москва, 1983 год

ej \ 37-01

Может ли путник выйти из лабиринта? Если может, то какой путь он должен пройти?

Дана прямоугольная карта лабиринта, в которой клетка равна 0, если она проходима, или 1, если непроходима. Путник может перемещаться в лабиринте из одной проходимой клетки в другую, если они имеют общую сторону. Путник выходит из лабиринта, когда попадает в граничную клетку.

Формат ввода:

m n

$a[1,1]$ $a[1,2]$... $a[1,n]$

$a[2,1]$ $a[2,2]$... $a[2,n]$

...

$a[m,1]$ $a[m,2]$... $a[m,n]$

k l

Здесь m, n – количество строк и столбцов в карте лабиринта ($1 \leq m, n \leq 80$); $a[i, j]$ – информация о лабиринте; k, l – исходное положение путника в проходимой клетке.

Формат вывода: первая строка: длина пути или слово "no", если выхода нет. Остальные строки: последовательность координат клеток, в которые необходимо переместиться, чтобы выбраться из лабиринта кратчайшим путём.

input.txt

```
9 10
1 1 1 1 1 0 1 1 1 1
1 0 0 0 0 0 1 1 1 1
1 0 1 1 1 1 1 1 0 0
1 0 0 0 0 0 1 1 0 1
1 1 1 0 1 1 0 0 0 1
1 1 1 0 1 1 0 1 1 1
1 1 0 0 0 0 0 0 0 1
1 1 1 1 0 1 1 1 0 1
1 1 1 1 1 1 1 1 1 1
7 6
```

output.txt

```
8
7 7
6 7
5 7
5 8
5 9
4 9
3 9
3 10
```

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 5 - 8 кл \ 10 - "Числа-подстроки"

dl.gsu.by \ Олимпиады по информатике

USACO \ 2005 \ April \ silver \ 2 - "Navigating the City"

Велосипед Jill

АСМ Финал

1995 год

ej \ 37-02

Jill Bates ненавидит подниматься в гору. Jill повсюду ездит на велосипеде, но она всегда хочет ехать по самой легкой и самой короткой дороге, насколько это возможно. Хорошие новости заключаются в том, что она живет в Greenhills, все дороги которого расположены в виде прямоугольной решетки: дороги "восток-запад" называются улицами, а дороги "север-юг" называются авеню и расстояние между двумя соседними точками решетки одно и то же. Плохие же новости заключаются в том, что Greenhills очень холмистый и в нём много односторонних дорог.

Выбирая дорогу от начального пункта к конечному Jill руководствуется тремя правилами:

1. Избегать подъём более чем на 10 метров между соседними точками решетки.
2. Никогда не ездить по односторонним дорогам в запрещённом направлении.
3. Всегда ехать по кратчайшему маршруту.

Ваша программа должна помочь Jill найти подходящий маршрут.

Формат ввода: Входной файл содержит данные в следующей форме:

Первая строка содержит 2 целых числа, разделённых одним или более пробелами. Первое число n представляет число улиц, а второе число m - содержит число авеню, $1 \leq n \leq 20$, $1 \leq m \leq 20$.

Следующие n строк содержат высоты точек решетки $h_{ij} \leq 40$. Каждая строка представляет улицу и содержит последовательность из m целых чисел, разделённых одним или более пробелами. Эти целые числа представляют высоту в метрах пунктов решетки вдоль улицы. Даже если какие-то улица и авеню не имеют пересечения, высота этой точки всё равно указывается.

Далее следуют одна или несколько строк, которые определяют односторонние дороги. Каждая дорога представляется двумя парами целых чисел, разделённых одним или более пробелами, в форме: улица авеню улица авеню. Первые улица и авеню определяют начальную точку дороги, а вторая пара определяет конечную точку дороги. Поскольку Greenhills - точная решетка, то в случае, когда две указанные точки не являются соседними в решетке, дорога проходит через все промежуточные точки решетки. Например, 5 7 5 10 представляет дороги из 5-7 в 5-8, из 5-8 в 5-9 и из 5-9 в 5-10. Определение дорог завершается строкой, содержащей 4 нуля в выше определённом формате.

Наконец, следуют одна или более строк (не более четырёх), которые содержат пары точек решетки, между которыми Jill хочет найти оптимальный путь, в форме: улица авеню улица авеню. Здесь, как и ранее, пары целых чисел разделены одним или более пробелами. Конец ввода данных определяется строкой, содержащей 4 нуля как описано ранее.

Вы можете рассчитывать, что все номера улиц и авеню будут находиться внутри границ, определённых первой строкой ввода, и что все дороги определены строго "север-юг" или "восток-запад".

Формат вывода: Для каждого запроса на путь во входном файле выведите последовательность точек решетки, от начальной точки решетки до конечной, представляющую путь, выбранный в соответствии с указанными правилами. Выводимые точки решетки указывайте как "улица-авеню" разделяя их словом "to".

Если не существует пути, удовлетворяющего критериям или если начальная и конечная точка одна и та же, выведите соответствующее текстовое сообщение. Выводите пустую строку между выходными наборами.

bike.in

```
3 4
10 15 20 25
19 30 35 30
10 19 26 20
1 1 1 4
2 1 2 4
3 4 3 3
3 3 1 3
1 4 3 4
2 4 2 1
1 1 2 1
0 0 0 0
1 1 2 2
2 3 2 3
2 2 1 1
0 0 0 0
```

bike.out

```
1-1 to 1-2 to 1-3 to 1-4 to 2-4 to 2-3 to 2-2

To get from 2-3 to 2-3, stay put!

There is no acceptable route from 2-2 to 1-1.
```

38. Рекурсия. Директива forward

dl.gsu.by \ Методы алгоритмизации

Задачи для книги \ Рекурсия \ P1 - "Сумма M из N"

Комментарий: следует учитывать случай, когда ни один из элементов не выбран

dl.gsu.by \ Методы алгоритмизации

Задачи для книги \ Рекурсия \ P2 - "Разложение числа в сумму"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2015 \ Школьная 1-11 кл, 7 октября \ 9 - 11 кл \ 7 - "Разложение в сумму различных"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Городская 1-9 кл, 11 апреля \ 8 - 9 кл \ 7 - "Количество способов"

ej

2017-09-03 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ Е. Сверхчеловек

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2001 \ Начинающие \ День 1 \ 3 - "Микросхемы-1"

dl.gsu.by \ Олимпиады по информатике
USACO \ 2014 \ February \ Bronze \ 3 - "SCODE_BRONZE"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 7 - "Любимые числа"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 9 - "Домино"
Комментарий: полагайте, что кости домино можно вращать

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинающие \ День 2 \ 1 - "Микросхемы-2"

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2000 \ Командные соревнования \ Командный тур (начинающие) \ G -
"Арифметическое выражение"

Каламбур

Автор: Алексей Ропан
ej \ 38-01

Определите, находится ли слово в таблице с буквами. Цепочка букв слова может изгибаться в таблице, но не может пересекаться сама с собой.

Формат ввода: В первой строке заданы числа: n и m – размер таблицы. $1 \leq n, m \leq 20$. Далее следуют n строк по m символов – таблица букв. На следующей строке после таблицы находится искомое слово, длина слова ≤ 10 .

Формат вывода: Выведите координаты начала слова на первой строке и координаты конца слова на второй строке.

Если подходящих слов несколько, выбирайте слово с меньшими координатами. Если слово не найдено, выведите "no".

input.txt

```
5 15
ABCDENGHIJAFLQV
KLFIROIRSTBGMRW
UVWXEZTBCDCHNSX
DEFGSTAKLMDJOTY
NOPQRSTUVWEKPUZ
FIRESTATION
```

output.txt

```
2 3
1 6
```

Трубопровод

Областная олимпиада школьников по информатике. День 1

Гомель, 1999 г.

Соавторы: Антон Малашенков, Иван Ревт

Компания Oil Transport Inc. решила провести трубопровод между двумя пунктами. В распоряжении компании имеются прямолинейные трубы различной длины и карта местности. Местность на карте разделена на квадратные участки единичного размера. На некоторых участках могут находиться болота, леса, реки и т.п. - здесь трубы уложить нельзя. Участки на карте расположены в N строк по M столбцов.

Трубы могут укладываться по горизонтали либо по вертикали, причём начало каждой следующей трубы совпадает с концом предыдущей. В каждом участке местности может быть проведено несколько труб.

Ваша программа должна помочь компании провести трубопровод от пункта А в пункт В.

Формат ввода: в первой строке входного файла содержатся числа N и M ($2 \leq N, M \leq 25$), разделённые пробелом. В следующих N строках содержится карта местности по M чисел в строке. Число 0 обозначает участок, на котором можно уложить трубы, а 1 - участок, на котором трубы уложить нельзя. В $N+2$ -й строке файла содержится число K - количество имеющихся труб ($1 \leq K \leq 7$), а в следующих K строках - их длины, не превышающие 25. В последней строке находятся числа A_x, A_y, B_x, B_y - координаты начального и конечного пунктов (A_x, B_x - номера строк, A_y, B_y - столбцов).

Формат вывода: в первой строке выходного файла находится единственное число P - количество использованных труб. В следующих P строках содержатся числа A_i, X_i и Y_i , разделённые пробелами, где A_i - номер i -й использованной трубы, а X_i, Y_i - координаты её конца (X_i - строка, Y_i - столбец). Если трубопровод проложить невозможно, выходной файл должен содержать единственную строку "NO SOLUTION".

input.txt

```
4 5
1 1 1 1 1
1 0 0 0 0
1 0 1 0 0
1 0 1 0 0
3
4
5
3
4 2 2 5
```

output.txt

```
2
3 2 2
1 2 5
```

Головоломка

Областная олимпиада школьников по информатике. День 2

Гомель, 1999 г.

Вашему вниманию предлагается головоломка. Она представляет собой равносторонний деревянный треугольник, в котором просверлены 15 отверстий. Изначально в некоторых отверстиях стоят фишки. Каждым ходом нужно "сбить" одну фишку. Фишка сбивается, когда через неё переступают другой фишкой. Переступать можно только через смежную фишку

(место, на которое становится фишка, должно быть свободно). Так как поле треугольное, то у каждого отверстия может быть до шести смежных отверстий. Ваша задача ходить таким образом, чтобы в результате на поле осталась только одна фишка. Все отверстия пронумерованы от 1 до 15, как это показано на рисунке.

Ваша задача - написать программу, которая для заданного начального расположения фишек определяет одну из выигрышных последовательностей ходов. Если же выигрышной последовательности ходов нет, то необходимо выдать сообщение "NO SOLUTION".

```

      1
     2 3
    4 5 6
   7 8 9 10
  11 12 13 14 15

```

Формат ввода: первая строка входного файла содержит число N - количество фишек в начальном расположении. Далее N строк с номерами отверстий с фишками.

Формат вывода: ваша программа должна вывести в выходной файл последовательность ходов в следующем формате:

```

A(1) B(1)
A(2) B(2)
...
A(N-1) B(N-1)

```

где A_i B_i определяют i -й ход (походили фишкой из позиции A_i в позицию B_i). Если же выигрышной последовательности нет, то выходной файл должен содержать только одну строку "NO SOLUTION".

Пример 1

input.txt

```

3
1
2
5

```

output.txt

```

1 4
4 6

```

Пример 2

input.txt

```

3
1
2
6

```

output.txt

```

NO SOLUTION

```

Пароль

Областная олимпиада по информатике, командный тур
Гомель, 11 января 2002 г

Вы - первоклассный хакер и работаете на федеральную службу безопасности (ФСБ). У вас секретная миссия - получить важную информацию из Пентагона. Но, к сожалению, данные заблокированы сильной защитной системой. Вы должны написать программу, решающую эту

проблему. Пароль к системе меняется каждый день, так что есть только один способ узнать его - получить специальный ключевой файл, в котором лежат кодированные данные.

Файл состоит из тонкой кроссвордной сетки, частично заполненной символами '#', и списка слов. Сетка уже содержит несколько горизонтальных слов. Столбцы пусты и должны быть заполнены некоторыми вертикальными словами. Вы должны вставить несколько слов из списка в пустые столбцы так, чтобы каждое вставленное слово пересекалось с горизонтальным в одной и той же букве. Каждое слово из списка может быть взято один раз, также могут остаться неиспользованные слова.

ФСБ известно из достоверных источников, что пароль может быть определён как последовательность первых букв, вставленных в пустые столбцы, отсчитывая столбцы слева направо. Если невозможно заполнить пустые столбцы описанным способом, тогда пароль должен быть назначен в слово 'impossible'.

Формат ввода: N - количество строк в сетке ($7 \leq N \leq 18$). M - количество столбцов в сетке ($6 \leq M \leq 91$). Далее следует сетка. T - количество слов в списке ($6 \leq T \leq 44$). Далее следуют слова по одному на строке ($2 \leq \text{длина слова} \leq 10$).

Формат вывода: пароль.

password.in

```
9
7
#####
####  ##
   ###  ##
      #   #
abv#dr#
   ##  #
      ##  #
#####  #
#####  #
7
kraja
ovod
sbeg
telenok
plata
svet
sboi
```

password.out

```
pssok
```

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 7 - "Сумма k подмножеств"

39. Быстрая сортировка. Объявление типов данных. Интерактивные задачи. Подключение библиотек

Быстрая сортировка наборов

Автор: Алексей Ропан

Даны наборы - количество чисел и массив чисел. Ввод наборов завершает 0. Нужно вывести каждый набор в файл, отсортировав в порядке невозрастания.

Формат ввода:

N_1 количество чисел ($1 \leq n \leq 16000$)
 $a_{1,1}$
 ... a - массив чисел ($-2\,147\,483\,648 \leq a_i \leq 2\,147\,483\,647$)
 $a_{1,n}$
 n_2
 $a_{2,1}$
 ...
 $a_{2,n}$
 ...
 0

Формат вывода: Отсортированные наборы по одному числу на строке.

sort.in	sort.out
4	5
3 4 5 1	4
1	3
2	1
0	2

Грузовики

Автор: Алексей Ропан

При создании нового искусственного водоема возникла проблема вывоза песка.

За то время, как грузовики вывозят песок и возвращаются обратно, выкапывается p тонн песка. Всего имеется n грузовиков с заданной грузоподъемностью.

Ваша задача определить, какое минимальное количество грузовиков понадобится для эффективного вывоза песка. Вывоз песка считается эффективным, если каждый раз вывозится весь накопившийся песок (этот способ всегда существует).

Формат ввода:

n p n - количество грузовиков ($1 \leq n \leq 30000$), p - количество песка ($1 \leq p \leq 5000000000$)
 a
 a_2 a_i - грузоподъемности грузовиков ($1 \leq a_i \leq 30000$)
 ...
 a_n

Формат вывода: k - минимальное количество грузовиков, необходимых для эффективного вывоза песка.

input.txt

3 10
5
6
4

output.txt

2

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2013 \ День 2 \ 2 - "Новогодняя елка"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2014 \ День 2 \ 2 - "Торговля"*

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2014 \ День 1 \ 4 - "Сейф - интерактивная"*

Комментарии:

- вызов `Try` после открытия сейфа неразрешён
- при отправке используйте компилятор GNU C++ 3.4.2

*codeforces.ru \ Тренировки
2013-2014 Цикл интернет-олимпиад. Вторая командная олимпиада, базовый уровень (6 октября 2013 года) \ F. Побег*

*codeforces.ru \ Тренировки
2013-2014 Цикл интернет-олимпиад. Вторая командная олимпиада, базовый уровень (6 октября 2013 года) \ G. Стражи антилопы и гиены*

*codeforces.ru \ Архив
840A. Леха и функция*

*codeforces.ru \ Тренировки
2013-2014 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов (ВКОШП 13, Интернет-тур) \ E. Разделение королевства*

*codeforces.ru \ Тренировки
2013-2014 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов (ВКОШП 13, Интернет-тур) \ A. Банк*

[codeforces.ru \ Архив](#)
[733D. Скульптор Костя](#)

[ej](#)
[2017-09-03 \[Дорешивание\] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ Н. Blink Dagger](#)

[codeforces.ru \ Архив](#)
[1282D. Заколдованный артефакт](#)

[official.contest.yandex.ru/ioi](#)
[IOI 2018 \ A. combo](#)

40. Жадные алгоритмы

[codeforces.ru \ Группы \ КЮП](#)
[ФПМИ 2015 \(отбор\) \ В. Монеты и гнезда](#)

[codeforces.ru \ Группы \ КЮП](#)
[ФПМИ 2015 \(отбор\) \ G. Рекламный плакат](#)

[codeforces.ru \ Архив](#)
[441B. Валера и фрукты](#)

[codeforces.ru \ Архив](#)
[525C. Илья и палочки](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[GCSW \ 2000 \ Личные соревнования \ Начинающие \ День 2 \ C - "Переписывание мультиков"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[USACO \ 2010 \ February \ Silver \ 1 - "Chocolate Buying"](#)

Yogurt Factory

[USACO March 2008 Silver division](#)
Автор тестов: Евгений Грицкевич

Коровы приобрели йогуртовую фабрику, которая производит всемирно известный Yucky Yogurt. В течение следующих N ($1 \leq N \leq 100000$) недель цены на молоко и труд будут меняться

еженедельно так, что одна единица йогурта в неделю i будет стоить C_i ($1 \leq C_i \leq 10000$) центов. Фабрика может производить бесконечно много йогурта каждую неделю.

Yucky Yogurt владеет складом, который может хранить неиспользованные йогурты по цене S ($1 \leq S \leq 100$) центов за единицу йогурта в неделю. На складе йогурт не портится; там может храниться неограниченное количество йогурта.

Yucky хочет делать поставки еженедельно по Y_i ($0 \leq Y_i \leq 10000$) единиц йогурта. Помогите Yucky минимизировать стоимость поставок за N -недельный период. Йогурт, произведённый в эту неделю, как и йогурт лежащий на складе, может быть использован для удовлетворения спроса.

Формат ввода:

N S
 C_1 Y_1
 ...
 C_n Y_n

Формат вывода: минимальная сумма, потраченная на производство йогуртов.

Комментарий:

В первую неделю производим 200 единиц, и сбываем всё.
 Во вторую неделю производим 700 единиц, 400 сбываем, 300 кладём на склад.
 В третью неделю сбываем 300 единиц, которые лежат на складе.
 В четвёртую неделю производим и сбываем 500 единиц.

yogurt.in

4 5
 88 200
 89 400
 97 300
 91 500

yogurt.out

126900

dl.gsu.by \ Олимпиады по информатике
 USACO \ 2009 \ December \ Silver \ 3 - "Selfish Grazing"

codeforces.ru \ Архив
 845C. Два телевизора

codeforces.ru \ Архив
 437C. Ребенок и игрушка

dl.gsu.by \ Олимпиады по информатике
 USACO \ 2013 \ December \ Silver \ 1 - "MSCHED"

codeforces.ru \ Архив
 534D. Рукопожатия

codeforces.ru \ Тренировки
2011-2012 Командный чемпионат школьников Санкт-Петербурга по программированию
(XIX) \ I. Гури

codeforces.ru \ Архив
540B. Школьные оценки

codeforces.ru \ Архив
533E. Исправление ошибок

informatics.mccme.ru
Задача №112774. Кладовка из Простоквашино

codeforces.ru \ Архив
976E. Неплохо сыграно!

41. Дихотомия. Трихотомия. Префиксные суммы. Префиксные суммы на матрице

Коровий словарь

USA Computing Olympiad

Фермер Джон учит коров произносить некоторые фразы на человеческом языке. Он записал каждую фразу, которую его коровы должны выучить в специальную книгу фраз - всего M ($1 \leq M \leq 1\,000$) фраз.

Когда Фермер Джон на каникулах, он может общаться со своими коровами только по телефону. По возвращении, его автоответчик содержит N записей ($1 \leq N \leq 10\,000$), многие из которых возможно от коров.

Многие из записей обрывались прежде, чем были закончены. Помогите Фермеру Джону определить, являются ли записанные сообщения фразами из его книги коровьих фраз (то есть, что они начинаются с фразы из книги).

Вам дается книга фраз и множество текстов записанных сообщений.

Подсчитайте количество записанных сообщений, которые начинаются с фразы из книги фраз. Полные фразы имеют длину от 1 до 60 символов, каждый из которых или буква (большая или маленькая), или точка, или запятая, или знак вопроса, или знак пробела. Во фразе нет начальных и конечных пробелов, а также дублированных пробелов внутри фразы.

Сравнение нужно проводить регистро-чувствительное. Фраза является префиксом самой себя.

Формат ввода:

Строка 1: два разделенных пробелом целых числа M и N
 Строки 2..M+1: фразы из книги фраз по одной в строке
 Строки M+2..M+N+1: фразы из автоответчика, по одной в строке

Формат вывода: одно целое число - количество сообщений автоответчика, которые являются полными префиксами фраз из книги фраз.

Комментарий к примеру ввода: три входных фразы (record, eels, bouncy) и четыре фразы на автоответчике (buy, helicopter, come back, cat).

Комментарий к примеру вывода: сообщения "I will not buy this rec" и "Do you want to come back" являются корректными префиксами фраз из книги фраз.

phrase.in

```
3 4
I will not buy this record, it is scratched.
My hovercraft is full of eels.
Do you want to come back to my place? Bouncy, bouncy.
I will not buy this rec
My helicopter is
Do you want to come back
I will not buy this cat.
```

phrase.out

```
2
```

[codeforces.ru \ Архив](#)
[816B. Карен и кофе](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Российская \(ком., Inet\) \ 2008 \ Усложненные \ Олимпиада 11 \(3 мая 2008\) \ 3 -](#)
["Последовательность Хэмминга"](#)

[codeforces.ru \ Тренировки](#)
[2012-2013 Тренировка СПбГУ В #12 Бинарный поиск, тернарный поиск \ D. Корень](#)
[кубического уравнения](#)

[codeforces.ru \ Тренировки](#)
[2012-2013 Тренировка СПбГУ В #12 Бинарный поиск, тернарный поиск \ С. Поляна дров](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[COCI \ 2014 \ coci_2013-14 \ 14_COCI6_Mar \ 3 - "KOCKICE"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[USACO \ 2014 \ February \ Bronze \ 2 - "AUTO"](#)
Комментарий: полагайте, что $N \leq 10\,000$

codeforces.ru \ Тренировки

2012-2013 Тренировка СПбГУ В #12 Бинарный поиск, тернарный поиск \ В. Коровы - в стойло

codeforces.ru \ Тренировки

2012-2013 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов (ВКОШП 12, Интернет-тур) \ А. Рекламное объявление

codeforces.ru \ Архив

778A. Игра на строке

ej

2018-05-27 Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ Н. Игра с массивом

dl.gsu.by \ Программирование - профессионалы (ком. 2008) (P/O)

Nee4n_2005 \ 4 - "Drying"

dl.gsu.by \ Олимпиады по информатике

Гомельская обл. \ 1997 \ D - "Голодание"

informatics.mccme.ru

Задача №111236. aliens

ej

2018-05-27 Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ Г. Оптимальный выбор

dl.gsu.by \ Олимпиады по информатике

Гомельская обл. \ 2014 \ День 1 \ 2 - "Удачная фотосессия"

42. Двоичная система счисления. Алгоритм генерации двоичных чисел (Двоичная система счисления. Алгоритм генерации двоичных чисел)

Двуландия

Авторы: Владислав Собин, Владимир Миранович

Учёные открыли новую планету под названием "Двуландия". Было обнаружено, что на этой планете живет раса, у которой имеют хождение свои денежные единицы. Одного из учёных

это так заинтересовало, что он около 5 лет копил деньги, чтобы организовать экспедицию на планету.

Учёному не хватает для отъезда n монет. Ему надо узнать, чему равна недостающая сумма в денежных единицах "Двуландии". Каждый разряд в числах на этой планете не больше 1 и не меньше 0. Каждый разряд имеет свою ценность: первый - 1, второй - 2, третий - 4 и так далее. Помогите учёному определить, сколько денег не хватает ему, чтобы полететь на планету "Двуландия".

Формат ввода: одна строка, содержащая число n . $1 \leq n \leq 2\,147\,483\,647$.

Формат вывода: одна строка, содержащая количество денег, которых не хватает учёному.

input.txt

100

output.txt

1100100

Двуландия 2

Автор: Владимир Миранович

Экспедиция подходит к концу, и ученому пора возвращаться на Землю. Но у него есть только деньги с планеты "Двуландия", а цена билета указана в земных денежных единицах.

Помогите ученому узнать, сколько у него денег в денежных единицах Земли.

Формат ввода: строка длиной n . $1 \leq n \leq 31$.

Формат вывода: количество денег у учёного.

input.txt

1100100

output.txt

100

codeforces.ru \ Apxue
465A. inc ARG

Единицы и нули

Автор: Владимир Миранович

Найдите все возможные варианты расположения 1 и 0 в строке длины n и выведите их согласно примеру вывода. $1 \leq n \leq 10$.

input.txt

2

output.txt

00

01

10

11

dl.gsu.by \ Олимпиады по информатике
Российская (ком., Inet) \ 2007 \ Базовые \ Олимпиада_6 (15 декабря 2007) \ 2 - "Хорошие цифры"

43. Операции сдвига. Быстрое умножение и деление на степень 2. Выделение части числа. Битовая маска. Быстрый остаток от деления на степень 2

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2010 \ День 2 \ 1 - "Бинарные близнецы"*

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2014 \ День 1 \ 1 - "День рождения"*

*dl.gsu.by \ Олимпиады по информатике
Московская \ 2013 \ С - "Жизнь на прямой"*

*dl.gsu.by \ Олимпиады по информатике
Канадская (Stage 1) \ 2011 \ Senior \ 5 - "Switch"*

*codeforces.ru \ Архив
405C. Необычное произведение*

*codeforces.ru \ Архив
424C. Волшебные формулы*

*dl.gsu.by \ Олимпиады по информатике
BalkanOI \ 2008 \ День 2 \ 1 - "The cure"*

44. Признаки степени 2. Перебор подмасок заданной маски. Быстрое возведение в степень. Вычисления по модулю. Код Грея

*e-olymp.com \ Задачи
273. Возведение в степень*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2016 \ День 1 \ 2 - "Технологии"*

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2012 \ День 2 \ 3 - "Конкурс"

dl.gsu.by \ Олимпиады по информатике
USACO \ 2005 \ April \ silver \ 3 - "Disease Management"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2017 \ Областная 1-9 кл, 20 апреля \ 5-9 кл \ 11 - "Игра с камешками"

45. Шестнадцатеричная система счисления. Перевод чисел между шестнадцатеричной и восьмеричной системами счисления. N-ричная система счисления

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинаящие \ День 2 \ 2 - "Штирлиц"

Комментарий: используйте последнюю цифру числа N в десятичном представлении.

codeforces.ru \ Архив
136B. Тернарная логика

Детские примеры

Автор: Владимир Миранович

Преподавательница по математике задала детям решить несколько примеров вида $a+b=c$, но, так как они уже давно проходили эту тему, то максимальная оценка, которую могут получить дети - не больше четырёх баллов. В классе был один отличник, и он не хотел, чтобы у него оценки были ниже пяти баллов. Он знал десятичную, двоичную и шестнадцатеричную системы счисления, и решил, что если узнает для каждой из трёх систем счисления, будет ли верно заданное выражение, то учительница оценит его знания и поставит пять баллов. Но, так как переводить числа из одной системы счисления в другую очень долго и сложно, он просит помощи у вас.

Формат ввода: не более $15 \cdot 10^3$ строк с примерами.

Формат вывода: на каждую строку выведите ответ согласно примеру вывода.

Комментарий: если сумму чисел верного выражения перевести в десятичную систему счисления, то она будет не больше, чем 2^{31} .

input.txt

10+1=11
2+2=4
A+a=14

output.txt

Answer# 1
10 - TRUE
2 - TRUE
16 - TRUE
Answer# 2


```

10 - TRUE
2 - FALSE
16 - TRUE
Answer# 3
10 - FALSE
2 - FALSE
16 - TRUE

```

codeforces.ru \ Архив
1249C2. Хорошие числа (сложная версия)

dl.gsu.by \ Олимпиады по информатике
USACO \ 2010 \ March \ Bronze \ 2 - "Hexadecimal Conversion"

46. Перестановки и сочетания

Перестановки

Автор тестов: Пётр Майоров

Прочитайте число n ($n < 9$) из файла, для последовательности чисел $1..n$ выведите все перестановки в словарном порядке.

input.txt

3

output.txt

```

1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

```

Цепочка лампочек

Автор тестов: Пётр Майоров

В цепочке из n ($n < 11$) лампочек должно быть одновременно включено m ($m \leq n$) из них. Для заданных n и m распечатайте в обратном лексикографическом порядке все возможные сочетания включенных лампочек в цепочке. 1 обозначает включенную лампочку, 0 – выключенную.

input.txt

5 3

output.txt

```

1 1 1 0 0
1 1 0 1 0
1 1 0 0 1
1 0 1 1 0
1 0 1 0 1
1 0 0 1 1

```

0	1	1	1	0
0	1	1	0	1
0	1	0	1	1
0	0	1	1	1

Перетягивание каната

Автор: Евгений Костюкевич

Для соревнований по перетягиванию каната в команде из 10 человек потребовалось отобрать такие группы по три человека, вес каждой из которых был бы равен 150 кг.

Напишите программу, которая из команды в n человек составляет всевозможные группы по m человек и печатает вес каждой полученной группы.

Формат ввода: в файле ввода прочитайте n , m и веса для n человек. $0 < m \leq n \leq 10$. Вес человека не превышает 100 кг.

Формат вывода: в файле вывода расположите веса групп в порядке их возрастания. Для групп, одинаковых по весу, печатайте вес только один раз.

rope.in	rope.out
5 3	135
40	145
60	150
100	155
45	185
50	190
	195
	200
	205
	210

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2008 \ День 2 \ 1 - "Криптомания"

dl.gsu.by \ Олимпиады по информатике
Санкт-Петер. (ком.) \ 2013 \ С - "cipher"

47. Признаки делимости. Наибольший общий делитель. Алгоритм Евклида. Наименьшее общее кратное. Числовые циклы. Период бесконечной дроби

dl.gsu.by \ Программирование – начинающие (P/O)
21-30 \ 3 \ 4 - "Делится ли число?"

Делимость

Автор: Евгений Грицкевич

Определите, делится ли число на 2, 3, 4, 5, 6, 8, 9, 10, 11, 12.

Формат ввода: строка с числом. Длина числа – до 70000 цифр.

Формат вывода: для каждого признака делимости "not divisible by" или "divisible by" согласно формату вывода.

delimost.in

363

delimost.out

not divisible by 2
divisible by 3
not divisible by 4
not divisible by 5
not divisible by 6
not divisible by 8
not divisible by 9
not divisible by 10
divisible by 11
not divisible by 12

codeforces.ru \ Архив
[822A. Мне скучно жить](#)

Натуральная дробь

Автор: Евгений Костюкевич

Прочитайте a и b . Приведите к натуральному виду дробь, числитель которой равен a , знаменатель – b . Если целая или дробная часть результата равна нулю, не выводите её. В случае, когда число равно нулю, выведите 0.

input.txt

60 24

output.txt

60/24=2+1/2

[ej](#)

[2017-07-09 \[Дорешивание\] Тренировочная олимпиада КЮП. Контекст Вячеслава Козловского и Ивана Ревта \ Н. Функция](#)

Шоколадка

Районная (городская) олимпиада по информатике 8-9 класс
Мозырь, ноябрь, 1998 г.

Как известно, реклама – двигатель торговли. При этом важно найти ту изюминку, которая заставила бы покупателя обратить внимание именно на ваш товар. Так одна кондитерская фабрика решила выпустить серию шоколадок под названием "По-братски". Их "изюминка"

заключается в том, что каждая шоколадка из этой серии состоит из такого количества кусочков, что её можно разделить поровну на любое количество частей, не больше какого-то n .

Шоколадки так и называются: "По-братски 2" – можно съесть одному, а можно поделить на двоих (2 кусочка); "По-братски 3" – можно съесть одному, поделить на двоих или на троих (6 кусочков); "По-братски 4" – 12 кусочков.

Фабрика решила не останавливаться на максимальном числе 4. Но, чем дальше, тем сложнее находить минимальное количество кусочков, необходимое для выполнения условия. Ваша задача – написать программу, которая поможет найти минимальное количество кусочков в шоколадке для заданного n .

Формат ввода: строки до конца файла, содержащие по одному числу n . $n \leq 22$.

Формат вывода: для каждой строки - количество кусочков.

<i>input.txt</i>	<i>output.txt</i>
1	1
2	2
3	6
4	12
5	60

codeforces.ru \ Архив
456B. Федя и математика

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2002 \ Начинающие \ День 2 \ 6 - "Математическая"

dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2002 \ День 1 \ 6 - "Смеси"

Таинственный предмет

Областная олимпиада по информатике, II тур
Гомель, 10 января 2002 г.

Археологи, проводя раскопки в Египте, натолкнулись на очень странный предмет. Выглядел он как небольшая коробочка, на которой находились таинственные знаки. Каждый день символы на коробочке менялись местами. Учёных заинтересовал этот предмет, и они провели некоторые исследования. Они заметили, что знаки всё время меняются по одному и тому же закону: символ, который стоит на i -ой позиции, каждый раз будет становиться на одно и то же место.

Для дальнейшего исследования ученые пронумеровали символы числами от 1 до n , где n - количество символов, и получили информацию, как меняются знаки. Ваша задача – определить, через сколько дней знаки возвращаются в начальное положение.

Формат ввода:

n количество знаков ($1 \leq n \leq 10000$).

$a_1 a_2 \dots a_n$ закон, по которому меняются символы

Например, запись 3 1 2 означает, что на место 1-го символа стал 3-й, на место 2-го - 1-й, на место 3-го - 2-ой.

Формат вывода: количество дней, через которое знаки вернутся в начальное положение. Гарантируется, что ответ не превышает 10^{12} .

device.in

10
10 9 8 7 6 5 4 3 2 1

device.out

2

Длина периода

Автор: Владимир Миранович

Вычислите длину периода бесконечной десятичной дроби, заданной выражением вида n / m .

Формат ввода: $n m$. ($1 \leq n, m \leq 30000$).

Формат вывода: длина периода.

Комментарий: в примере ввода $1/990=0.0010101\dots=0.0(01)$. Длина периода равна 2.

input.txt

1/990

output.txt

2

Бесконечная дробь

Автор: Сергей Коноплич

Вычислите значение десятичной дроби, заданной выражением вида n / m .

Формат ввода: $n m$. $n \leq \text{maxlongint}$. $m < 10000$

Формат вывода: значение дроби с периодом, заключенным в скобки, в формате, приведённом в примере вывода. Если период равен 0, не выводите его.

decimal.in

1 990

decimal.out

1/990=0.0(01)

informatics.mccme.ru

Задача №112089. Пляжный волейбол

48. Простые и составные числа. Алгоритмы определения простого числа. Решето Эратосфена. Двойное решето Эратосфена. Факторизация числа. Тернарная и бинарная гипотеза Гольдбаха

Easy Prime Numbers

Автор: Сергей Коноплич

Прочитайте n , выведите в файл все простые числа на интервале $1..n$. $1 \leq n \leq 60000$.

input.txt

20

output.txt

2
3
5
7
11
13
17
19

dl.gsu.by \ Олимпиады по информатике

GCSW \ 2003 \ Начинающие, ком. \ 2 - "Простые числа"

Комментарий: полагайте, что $a[i] \leq 1010$

Вася Пупкин ушёл в математику

Автор: Сергей Коноплич

Соавтор тестов: Антон Надёжкин

Вася Пупкин решил подкачаться в математике. Он поставил перед собой задачу – узнать, сколько делителей у числа n ($0 < n < 2\,000\,000\,001$).

Формат ввода: одна или несколько строк с числами для проверки. Количество строк не больше 5001.

Формат вывода: соответствующее файлу ввода количество строк с ответами: "One" — если делитель один, "Two" — если делителей два, "Many" — если делителей больше двух.

pupkin.in

13
1

pupkin.out

Two
One

Prime Numbers

Автор: Сергей Коноплич

Прочитайте n , выведите в файл все простые числа на интервале $1..n$. $1 \leq n \leq 140000$.

input.txt

output.txt

20

2
3
5
7
11
13
17
19

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинающие \ День 1 \ 4 - "Хакеры"

spoj.com \ problems
11. Factorial

codeforces.ru \ Архив
1114C. Замыкающие нули

ej
2017-06-18 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра
Кривошеева \ G. Саша и интересный массив

spoj.com \ problems
2. Prime Generator

informatics.mccme.ru
Задача №152. Гипотеза Гольдбаха

codeforces.ru \ Архив
735D. Налоги

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2013 \ День 1 \ 2 - "Почтальоны"

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2009 \ День 1 \ 1 - "Красивая стена"

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2012 \ День 1 \ 1 - "Сопряженный делитель"

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2008 \ December \ Silver \ 2 - "Patting Heads"*

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2014 \ День 2 \ 2 - "Квадрокуб"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2014 \ День 1 \ 4 - "Урок математики"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2015 \ День 2 \ 2 - "Куб в прогрессии"*

*codeforces.ru \ Тренировки
2013-2014 Всероссийская командная олимпиада школьников по программированию,
интернет-тур + отборы регионов (ВКОШП 13, Интернет-тур) \ F. Загадочное уравнение*

*codeforces.ru \ Тренировки
2008-2009 Цикл интернет-олимпиад. Шестая командная олимпиада (29 ноября 2008).
Усложненный уровень \ G. Точная степень*

*ej
2017-08-13 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра
Кривошеева \ Н. Поиск простого числа*

*codeforces.ru \ Архив
1029F. Разноцветные маркеры*

49. Контейнеры set, map. Итераторы

*ej
2017-06-18 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра
Кривошеева \ I. Путешествие Юры*

*codeforces.ru \ Архив
858D. Телефонная книга Поликарпа*

codeforces.ru \ Тренировки

2012-2013 Цикл интернет-олимпиад. Вторая командная олимпиада (20 октября 2012).

Базовый уровень. \ В. Автомобили в Байтландии

dl.gsu.by \ Олимпиады по информатике

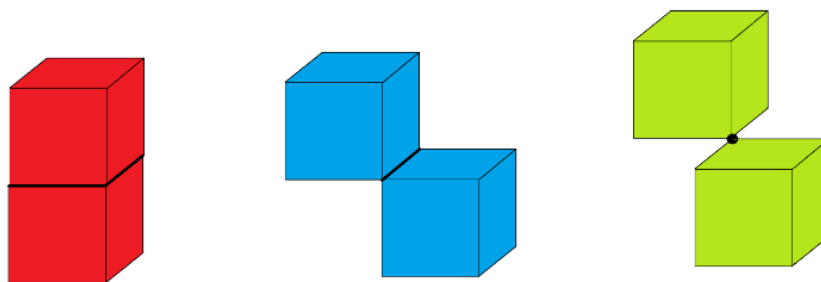
USACO \ 2013 \ January \ Silver \ 3 - "INVITE"

Кубическая вселенная

Автор: Руслан Лихач

Илья, Игорь и Андрей очень любят путешествовать по просторам космоса. На этот раз они на своём корабле прилетели в очень интересную вселенную. Эта вселенная представляет собой трёхмерное пространство, в котором всё имеет кубическую форму.

Каждый куб имеет свои координаты (x, y, z) , а также может являться либо пустым пространством, либо принадлежать какому-нибудь космическому телу (звезде, планете и т. д.). Космическое тело – это совокупность из непустых кубов, где каждый куб имеет общую грань, ребро или вершину с хотя бы одним другим кубом этого тела (см. рисунок).



Ребятам известно количество непустых кубов и их координаты в пространстве. Теперь они задались множеством вопросов: Илья хочет узнать количество космических тел, Игорь – размер самого большого из них, а Андрей – размер самого маленького из них. Помогите им в этом разобраться.

Формат ввода:

Вам дано n ($0 \leq n \leq 50\,000$) – количество непустых кубов. На следующих n строках даны координаты этих кубов ($-10^{18} \leq x, y, z \leq 10^{18}$).

Формат вывода:

На первой строке выходного файла выведите количество космических тел, на второй строке – размер наибольшего тела, а на третьей – наименьшего.

Пример 1

space.in

```
8
1 1 1
26 26 26
2 2 2
11 16 12
1 2 1
1 2 2
10 15 11
1 1 2
```

Пример 2**space.in**

```
3
345 11 58
346 12 59
347 11 58
```

space.out

```
3
5
1
```

space.out

```
1
3
3
```

[codeforces.ru \ Архив](#)
1154E. Две команды

[codeforces.ru \ Архив](#)
982D. Акула

[codeforces.ru \ Архив](#)
799E. Украшения для аквариума

50. Метод двух указателей. Метод двух указателей на матрице

ej

2017-07-09 [Дорешивание] Тренировочная олимпиада КЮП. Контекст Вячеслава Козловского и Ивана Ревта \ I. Строка для Оранжевого

[codeforces.ru \ Архив](#)
716B. Дополни слово

[dl.gsu.by \ Олимпиады по информатике](#)
Гомельская обл. \ 2009 \ День 2 \ 2 - "Защищенный пароль"

[dl.gsu.by \ Олимпиады по информатике](#)
Гомельская обл. \ 2012 \ День 1 \ 2 - "Что нам стоит дом построить!"

dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2013 \ День 1 \ 3 - "Матрица"

codeforces.ru \ Архив
814С. Пылкий поток любви

codeforces.ru \ Архив
733С. Эпидемия в Монстрополисе

51. Рекуррентные соотношения. Динамическое программирование. Алгоритмы собирания суммы. Задача о рюкзаке. Восстановление пути в динамическом программировании

[ej](#)
2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контеcт Андрея Мищенко и Игоря Макеенко \ F. Программа обучения

codeforces.ru \ Тренировки
2012-2013 Тренировка СПбГУ В #1 \ К. Лесенка

Путешественник

Областная олимпиада школьников по информатике
Гомель, 8-11 января 2001 г.

Однажды путешественник решил спланировать свою поездку из города a в город b . Он знает, что можно воспользоваться поездом, автобусом и маршрутным такси.

На поезде путешественник сможет доехать из города a в город b за n остановок поезда. Это — наиболее быстрый маршрут, но ещё между остановками поезда ходят автобусы и маршрутные такси.

Между остановками поезда расположено по k остановок для автобусов и маршрутных такси, причём автобус перевозит пассажиров с любой остановки на следующую остановку, а маршрутное такси с любой остановки — через одну остановку.

Задача состоит в том, чтобы посчитать количество всех возможных маршрутов путешественника.

Формат ввода: первая строка: два числа через пробел — n и k . $1 \leq n \leq 9$. $1 \leq k \leq 7$. n, k - целые числа.

Формат вывода: число возможных маршрутов путешественника.

Комментарий: схема для примера ввода:

(A) -x-x- (x) -x-x- (B)

travel.in

2 3

travel.out

20

Сберкасса

XIV областная олимпиада по информатике
Могилев, 3-6 января 2001 г.

В сберкассе имеется неограниченный набор купюр достоинством k_1, k_2, \dots, k_m рублей. Можно ли выдать деньги вкладчику, пожелавшему снять со своего счета n рублей?

Формат ввода: в первой строке файла записано натуральное число m ($m < 15$). Во второй строке содержатся m натуральных чисел - достоинства купюр ($1 \leq k_i \leq 50000, i = 1..m$). И в третьей строке указан размер вклада n ($0 < n < 1000000$).

Формат вывода: в первой строке этого файла должно быть записано одно слово "Yes" или "No" в зависимости от того, возможна ли выплата вклада или нет.

acc.in

3
2 5 7
17

acc.out

Yes

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2003 \ День 1 \ 6 - "Числа"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 8 - "Рюкзак"*

Марки

АСМ финал
1995 год

Филателисты коллекционируют марки и тем самым доставляют неудобства почтовым работникам. Избыток марок может быть плохой новостью для почтовой службы страны, но хорошей новостью для тех, кто коллекционирует лишние марки. Почтовая служба старается минимизировать количество марок, необходимых для обеспечения нужд почты. Поэтому Вас попросили написать программу, которая поможет почтовой службе.

Размер конверта ограничивает количество марок, которое может быть использовано на одном конверте. Например, если 1-центовая и 3-центовая марки доступны и на конверт может быть наклеено 5 марок, все почтовые цены от 1 до 13 центов могут быть покрыты.

Цена	Марок по 1 центу	Марок по 3 цента
1	1	0

2	2	0
3	0	1
4	1	1
5	2	1
6	0	2
7	1	2
8	2	2
9	0	3
10	1	3
11	2	3
12	0	4
13	1	4

Хотя 5 3-центовых марок обеспечат конверт с 15-центовой стоимостью, невозможно обеспечить 14-центовую стоимость марок на конверте используя только 1- и 3-центовые марки (общим количеством не более 5). Поскольку почтовое ведомство желает иметь максимальное покрытие без пропусков, таковым в данном случае считается 13 центов.

Формат ввода: первая строка каждого набора данных содержит целое число s , представляющее максимальное число марок, размещающихся на конверте.

Вторая строка содержит целое число n , представляющее количество наборов номинаций марок в этом наборе данных.

Каждая из следующих n строк содержит набор номинаций марок. Первое целое число на каждой строке - количество номинаций в наборе, за ним следует список номинаций марок в порядке от меньшего значения к большему, причём каждое следующее значение номинации марки от предыдущего отделяется одним или несколькими пробелами. На каждой из этих n строк будет не более s номинаций.

Максимальная величина s - 10, максимальное значение номинации марки - 100, максимальное значение n - 10. Ввод ограничивается набором данных, начинающимся с 0 ($s=0$). Максимальное количество наборов данных: 3.

Формат вывода: вывод содержит по одной строке для каждого набора данных, давая максимальное покрытие без пропусков заданными номинациями марок в следующем формате:

max coverage = <число> : <список номинаций через пробел>

Если более чем один набор номинаций обеспечивает одно и то же максимальное число для покрытия без пропусков, нужно вывести набор, в котором меньше количество номинаций (это позволяет сократить расходы на изготовление марок).

Если есть несколько наборов, которые имеют одинаковые количества номинаций, то выбирается тот набор, в котором максимальная номинация имеет меньшее значение по сравнению с максимальными номинациями других наборов.

Например, если на конверте размещается 5 марок, то наборы марок 1, 4, 12, 21 и 1, 5, 12, 28 оба обеспечивают максимальное покрытие 71 цент. Но в качестве правильного ответа должен быть принят первый набор, поскольку наибольшая номинация в первом наборе (21) меньше наибольшей номинации во втором наборе (28).

stamps.in

stamps.out

5	max coverage = 71 : 1 4 12 21
2	max coverage = 409 : 1 7 16 31 88
4 1 4 12 21	max coverage = 48 : 1 5 7 8
4 1 5 12 28	
10	
2	
5 1 7 16 31 88	
5 1 15 52 67 99	
6	
2	
3 1 5 8	
4 1 5 7 8	
0	

dl.gsu.by \ Олимпиады по информатике

[USACO \ 2009 \ March \ Silver \ 2 - "Cow Frisbee Team"](#)

Комментарий: ограничения для входных данных смотрите в английском варианте условия

informatics.mccme.ru

Задача №1793. Наибольшая возрастающая подпоследовательность за $O(n \cdot \log(n))$

ej

2017-09-03 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева и Сергея Пискуна \ I. Торговец

codeforces.ru \ Тренировки

2012-2013 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов (ВКОШП 12, Интернет-тур) \ Е. Половина

codeforces.ru \ Архив

710E. Генерация строки

codeforces.ru \ Архив

455A. Скука

codeforces.ru \ Архив

583D. И снова ...

codeforces.ru \ Архив

463D. Gargari и перестановки

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2009 \ День 1 \ 2 - "Сладкие забавы"*

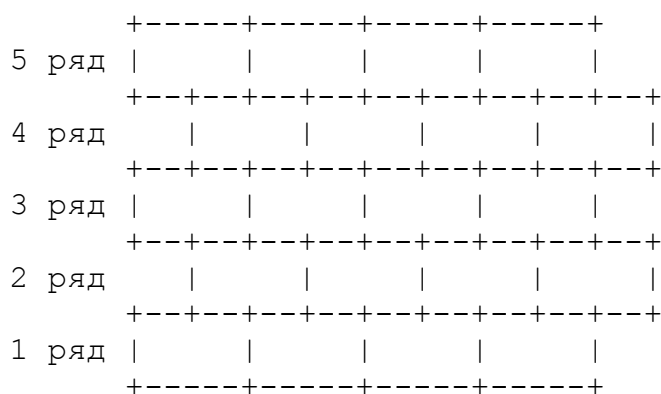
*dl.gsu.by \ Олимпиады по информатике
USACO \ 2007 \ February \ Silver \ 1 - "The Cow Lexicon"*

52. Двумерные рекуррентные соотношения

Кирпичная кладка

Областная олимпиада школьников по информатике
Гомель, 8-11 января 2001 г.

Кирпичная стена состоит из n рядов по k кирпичей, как показано на рисунке. В результате дефектоскопического исследования было выяснено, какую нагрузку (удар) выдерживает каждый кирпич.



Стена рухнет, когда по ней проходит сплошная трещина от верхнего ряда до нижнего. При этом нагрузка на стену должна быть не меньше, чем сумма нагрузок на разрушенные кирпичи. Требуется выяснить, какова наименьшая нагрузка, при которой стена разрушится.

Ряды нумеруются снизу вверх, как показано на рисунке. Кирпичи в каждом ряду нумеруются слева направо. При возникновении трещины разрушается ровно один кирпич в каждом ряду, а трещина проходит через соприкасающиеся кирпичи. $1 \leq$ количество рядов ≤ 100 . $1 \leq$ количество кирпичей в ряду ≤ 100 . $0 \leq$ сила удара, которую выдерживает один кирпич ≤ 50 .

Формат ввода:

n количество рядов

k количество кирпичей в каждом ряду

$u_{1,1}$

... $u_{i,j}$ - удар, который выдерживает j -тый кирпич в i -том ряду

$u_{1,k}$

$u_{2,1}$

...

$u_{2,k}$

...

$u_{n,1}$

...

 $u_{n,k}$

Формат вывода: наименьшая нагрузка, при которой стена разрушится.

Комментарий: рисунок для данного примера:

```

+--+--+--+--+--+--+--+--+--+--+
|  3  |  2  |  4  |<<3>>|
+--+--+--+--+--+--+--+--+--+--+
      |  5  |  4  |<<1>>|  4  |
+--+--+--+--+--+--+--+--+--+--+
|  6  |  5  |  8  |<<6>>|
+-----+-----+-----+-----+

```

Трещина пройдет по кирпичам, которые выделены на рисунке

brick.in

3
4
6
5
8
6
5
4
1
4
3
2
4
3

brick.out

10

codeforces.ru \ Архив
1207С. Газопровод

dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2002 \ День 2 \ 6 - "Робот"

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2012 \ Областная 1-9 кл, 4 мая \ 8 - 9 кл \ 8 - "Игра с пыльной полки"

Слова

XIV областная олимпиада по информатике
Могилев, 3-6 января 2001 г.

Мама начала обучать маленького Вовочку азбуке. Однако непоседливый Вовочка из всего алфавита запомнил только три буквы. Поскольку мама говорила, что из букв складываются

слова, то Вовочка решил продемонстрировать приобретенные навыки на расположенном перед домом дощатом заборе.

Из эстетических соображений на каждой доске записывалась только одна буква. Всего на заборе было n досок. Так как правил составления слов Вовочка не знал, то он решил, что словом может быть любой набор известных ему букв, которые не должны повторяться более чем k раз подряд.

Вовочка знал, что длинные слова нравились маме гораздо больше, чем короткие, поэтому он составлял слова во всю длину забора.

Определите, какое количество слов сумел написать Вовочка.

Формат ввода:

n
 k
 $0 < n < 35$
 $0 < k < 10$

Формат вывода: строка с искомым числом.

words.in

3
4

words.out

27

[codeforces.ru \ Архив](#)
[478D. Красно-зеленые башни](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 8 - "КТ-числа-2"](#)

[dl.gsu.by \ Олимпиады по информатике](#)
[USACO \ 2008 \ February \ Silver \ 3 - "Eating Together"](#)

[codeforces.ru \ Тренировки](#)
[2011-2012 Командный чемпионат школьников Санкт-Петербурга по программированию \(XIX\) \ D. Пизанская башня](#)

[codeforces.ru \ Архив](#)
[38E. Ну-ка, покатались!](#)

[codeforces.ru \ Архив](#)
[1132F. Удали строку](#)

codeforces.ru \ Тренировки

2018-2019 Всероссийская командная олимпиада школьников по программированию, интернет-тур + отборы регионов (ВКОШП 18, интернет-тур) \ С. Как проиграть контекст

ej

2018-05-27 Тренировочная олимпиада для начинающих. Контекст Александра Кривошеева \ J. Саша и мячики

Индекс слова

ACM International Collegiate Programming Contest

Sponsored by Microsoft

East-Central Regionals, 1995

Соавтор тестов: Валерий Камеко

Схемы кодирования часто используются в ситуациях, требующих кодирования информации или экономии при хранении/передаче. Здесь мы разработали простую схему кодирования, которая кодирует частично типы слов с пятнадцатью или меньше буквами нижнего регистра как целые числа.

Рассмотрим английский алфавит $\{a,b,c,\dots,z\}$. При использовании этого алфавита набор верных слов должен быть сформирован в строгом лексикографическом порядке. В этом наборе верных слов последовательные буквы слова находятся в строго возрастающем порядке, это значит, что следующие буквы в верном слове всегда находятся после предыдущих букв в соответствии с их положением в алфавите. Например, abc aер gwz верные трёхбуквенные слова, в то время как aab are cat - нет.

Для каждого верного слова ассоциируйте целое число, которое даёт позицию слова в алфавитном списке слов. То есть:

a	1
b	2
...	
z	26
ab	27
ac	28
...	
az	51
bc	52
...	
vwx _{yz}	83681
...	

Ваша программа должна прочитать серии входных строк. Каждая строка ввода будет иметь одиночное слово, которое будет иметь длину от одной до пятнадцати букв. Для каждого прочитанного слова выведите число 0, если слово неверно. Если прочитанное слово верно, выведите его индекс позиции в алфавитном списке.

Формат ввода: ввод состоит из серий одиночных слов, по одному на строке. Слова, по крайней мере, длиной в одну букву и не больше, чем в пятнадцать букв. Только алфавитные символы нижнего регистра $\{a,b,\dots,z\}$ будут использованы во вводе. Первая буква слова будет

первым символом на строке ввода. Ввод будет прерван концом файла. Количество строк в файле не превышает 100 000.

Формат вывода: выведите одиночное число, больше или равное нулю. Первая цифра выходного значения должна быть первым символом строки.

<i>index.in</i>	<i>index.out</i>
z	26
a	1
cat	0
vwxyz	83681

53. Элементы графа. Виды графов. Матрица смежности

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Школьная 1-11 кл, 5 октября \ 1 - 4 кл \ 20 - "Задача 20"

Архипелаг

Автор: Сергей Коноплич

Путешественник путешествует по одному из архипелагов. Между некоторыми его островами есть двусторонние дороги. Путешественник хочет узнать, доедет ли он от острова *a* до острова *b* по дороге.

Формат ввода:

m n количество городов и дорог. $0 < n, m < 100$

Следующие *n* строк содержат описание дорог.

Последняя строка содержит номера городов *a* и *b*.

Формат вывода: "yes", если путешественник доберётся из города *a* в город *b*, иначе - "no".

<i>input.txt</i>	<i>output.txt</i>
7 6	no
1 2	
2 3	
1 3	
3 4	
5 6	
6 7	
1 7	

Задана система дорог

Автор: Ионас Пакальнишкис, Александр Дряпко

Задана система городов и двусторонних дорог. Определите номера городов, недостижимых из города *a*.

Формат ввода: первая строка: $m\ n$ – количество городов и дорог. Строки 2.. $(n+1)$: описание дорог. Строка $(n+2)$: a . $m, n \leq 1000$

Формат вывода: города, недостижимые из a , расположенные в порядке возрастания номеров.

input.txt

```
7 6
1 2
2 3
1 3
3 4
5 6
6 7
1
```

output.txt

```
5 6 7
```

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2003 \ День 2 \ 5 - "Праздник"*

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2009 \ October \ Gold \ 5 - "The Leisurely Stroll"*

ej

2017-08-20 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Андрея Мищенко и Игоря Макеенко \ Е. Саша и путешествие

ej

2017-07-09 [Дорешивание] Тренировочная олимпиада КЮП. Контест Вячеслава Козловского и Ивана Ревта \ D. Славко и водопровод

Кратчайший путь

Автор: Адам Бардашев

Задана система городов и дорог. Определите кратчайший путь из города a в город b .

Формат ввода: первая строка: $m\ n$ – количество городов и дорог. Строки 2.. $(n+1)$: описание дорог. Строка $(n+2)$: $a\ b$. $m \leq 10$

Формат вывода: слово "no" или путь в порядке перемещения по городам включая начальный город.

input.txt

```
10 12
1 2
1 3
3 4
2 4
```

output.txt

```
1 2 5 7
```

```

2 5
4 5
4 6
6 7
5 7
8 9
9 10
8 10
1 7

```

Путешествие

Областная олимпиада по информатике, командный тур
Гомель, 11 января 2002 г.

У путешественника есть карта, где несколько городов соединено дорогами между собой. Ему требуется проехать из одного города в другой. Однако, этот путешественник очень осторожен, перед тем, как совершить свою поездку, он хочет узнать все возможные пути, которыми он может переместиться между городами. Ему не нравится посещать один и тот же город дважды. Ваша задача - помочь ему.

Задано общее количество городов P ($0 \leq P \leq 10$), описания дорог, связывающих города, и индексы исходного и конечного городов M и N . Необходимо найти все возможные маршруты между городами M и N .

Замечание: если $P=1$, тогда существует только один маршрут с длиной 1.

Формат ввода:

```

P
F[1,1] F[1,2] ... F[1,P]
F[2,1] F[2,2] ... F[2,P]
...
F[P,1] F[P,2] ... F[P,P]
M N

```

Здесь P - это количество городов. $F[i,j]$ - это логическое значение, описывающее существование дороги между городами i и j . M - индекс исходного города. N - индекс конечного города.

Формат вывода:

```

L[1 1] L[1 2] ... L[1 K]
L[2 1] L[2 2] ... L[2 K]
...
L[D 1] L[D 2] ... L[D K]

```

Здесь $L[i,j]$ - индекс j -го города в последовательности i -го маршрута.

Замечание: карта дорог может быть представлена как двумерный массив, где элемент $[i,j]$ равен 1, если существует дорога между i -м и j -м городами, и 0 в противном случае.

trip.in

```

7
0 1 1 1 0 0 0

```

trip.out

```

2 1 3
2 1 4 3

```

```

1 0 0 0 0 0 0
1 0 0 1 0 0 1
1 0 1 0 0 1 0
0 0 0 0 0 1 1
0 0 0 1 1 0 1
0 0 1 0 1 1 0
2 3

```

```

2 1 4 6 7 3
2 1 4 6 5 7 3

```

[codeforces.ru \ Архив](https://codeforces.ru/archive/problem/445B)
 445B. DZY любит химию

54. Список связности. Алгоритм Флойда. Индексирование входных данных

[codeforces.ru \ Группы \ КЮП](https://codeforces.ru/group/KYOP)
 Сборы #2 \ Е. Деревни

[codeforces.ru \ Архив](https://codeforces.ru/archive/problem/987D)
 987D. Ярмарка

informatics.mccme.ru
 Задача №112090. Турист Петр

ej
 2018-05-27 Тренировочная олимпиада для начинающих. Контеcт Александра Кривошеева \
 Е. Интересное путешествие

[codeforces.ru \ Архив](https://codeforces.ru/archive/problem/1204C)
 1204C. Анна, Святослав и карта

Best way

Автор: Алексей Синкевич

Задана система городов и дорог. Определите кратчайший путь из города a в город b .

Формат ввода: первая строка: m n – количество городов и дорог. Строки 2.. $(n+1)$: описание дорог в формате: начало дороги, конец дороги, длина дороги. Строка $(n+2)$: a b . $m \leq 10$

Формат вывода: путь в порядке перемещения по городам или "no".

Пример 1

input.txt

output.txt

```

5 6
1 2 5
1 5 100
2 4 10
2 3 1
3 4 2
5 4 3
1 5

```

```

1 2 3 4 5

```

Пример 2**input.txt**

```

5 6
1 2 5
1 5 100
2 4 10
2 3 1
3 4 2
5 4 3
1 1

```

output.txt

```

1

```

Кладоискатель

Автор: Алексей Ропан

Однажды кладоискатель нашёл в пещере интересный мини-город. В этом городе каждая дорога имела свой цвет. На входе висела табличка с надписью: "Если хочешь заработать m золотых, ты должен пройти из района a в район b , двигаясь только по дорогам цвета c ".

Кладоискатель не хочет долго искать дороги из района в район, поэтому он просит вас о помощи. Он составил список районов, соединённых между собой дорогами определённого цвета. Помогите подсчитать максимальное число золотых, которое может получить кладоискатель.

Формат ввода:

n количество дорог ($0 \leq n \leq 100$)

$x_1 y_1 z_1$
 $x_2 y_2 z_2$ дорога цвета z_i между x_i -ым и y_i -ым районами
 ... ($1 \leq x_i, y_i \leq 100$, $1 \leq z_i \leq 255$)

$x_n y_n z_n$

k количество наборов ($0 \leq k \leq 1000$)

$a_1 b_1 c_1 m_1$
 $a_2 b_2 c_2 m_2$ $a_i b_i$ - районы, c_i - цвет, m_i - количество золотых
 ... ($1 \leq a_i, b_i \leq 100$, $1 \leq c_i \leq 255$, $1 \leq m_i \leq 30000000$)

$a_k b_k c_k m_k$

Формат вывода: выведите максимальное количество золотых, которое может получить кладоискатель.

input.txt

```

9
1 2 1
1 4 2

```

output.txt

```

801

```

```

1 4 3
2 3 1
2 3 2
2 3 2
2 4 2
2 4 3
3 4 1
4
1 2 3 300
1 4 1 500
4 1 5 100
2 2 100 1

```

WWW

АСМ Финал
2000 год

Известно, что для того, чтобы перейти от любой страницы в World Wide Web к любой другой странице, надо сделать в среднем 19 кликов. То есть, если эти страницы рассматривать как вершины в графе, то среднее расстояние между произвольными парами узлов в графе - 19.

Дан граф, в котором все узлы могут быть достигнуты от любой отправной точки, ваша задача найти самое короткое среднее расстояние между произвольными парами вершин. Например, рассмотрим следующий граф. Обратите внимание, что связи показываются, как направленные рёбра, если есть связь от страницы a к странице b , то это не означает связь от страницы b к странице a .

```

( 1 ) -----> ( 2 )
  ^ |             |
  | |             |
  | v             v
( 3 ) <----- ( 4 )

```

Длина самого короткого пути от вершины 1 к вершинам 2, 3 и 4 равна 1, 1 и 2 соответственно. От вершины 2 к вершинам 1, 3 и 4 самые короткие пути имеют длины 3, 2 и 1. От вершины 3 к вершинам 1, 2 и 4, самые короткие пути имеют длины 1, 2 и 3. Наконец, от вершины 4 к вершинам 1, 2 и 3 самые короткие пути имеют длины 2, 3 и 1. Сумма всех длин путей $1+1+2+3+2+1+1+2+3+2+3+1=22$. Так как есть 12 возможных пар вершин, то средняя длина пути равна $22/12$, или 1.833 (три цифры после запятой).

Формат ввода: входные данные содержат множество наборов тестов – не более, чем 100. Каждый набор состоит из произвольного числа пар целых чисел a и b , каждая пара обозначает связь от страницы с номером a к странице с номером b . Номера страниц будут всегда располагаться в диапазоне от 1 до 100.

Ввод для каждого набора будет закончен парой нулей. Дополнительная пара нулей будет следовать за последним набором. Граф не будет иметь ребра из себя в себя, и по крайней мере один путь будет существовать от каждой вершины в графе к каждой другой вершине в графе.

Формат вывода: для каждого набора определите среднюю длину самых коротких путей между каждой парой вершин с точностью до трёх знаков после запятой. Покажите эту длину и идентификатор теста (они пронумерованы последовательно начиная с 1) и сформируйте выход как показано в примере.

Комментарий: в файле вывода предложение "Case #: average length between pages = #.### clicks" печатается в одну строку. Здесь # - произвольная цифра.

page.in

```
1 2 2 4 1 3 3 1 4 3 0 0
1 2 1 4 4 2 2 7 7 1 0 0
0 0
```

page.out

```
Case 1: average length between pages = 1.833 clicks
Case 2: average length between pages = 1.750 clicks
```

Автоугон

Гомельская городская (районная) олимпиада школьников по информатике

9-11 класс

30 ноября 2002 г.

Автор тестов: Максим Старовойтов

Мальчик Петя решил покататься на машине. Так как своей машины у него еще не было, он решил взять ее на время у своего соседа, дяди Бори. Дождавшись, пока тот уйдет на работу, Петя сел в машину и поехал в свою любимую песочницу, расположенную на другом конце города. Однако Петя не учел, что в городе есть добрые дяди милиционеры, которые через некоторое время начали проявлять к его поездке определенный интерес. Когда он увидел, что за ним гонятся милицейские машины, Пете стало интересно, смогут ли его поймать, если он будет ехать к своей песочнице по кратчайшему пути.

Гарантируется, что кратчайший путь от дома Пети до песочницы единственен. Помогите Пете решить эту задачу, так как он занят ездой и не должен отвлекаться. Выведите номер перекрестка, в котором задержат Петю, или -1, если он все же доедет до песочницы.

Милиционер может задержать машину только если он приедет на перекресток, через который будет проезжать нарушитель, в то же самое время или раньше. Поскольку милиционеры любят быструю езду, они всегда едут на максимальной скорости. Город состоит из n ($2 \leq n \leq 100$) перекрестков и m соединяющих их дорог, длина каждой из которых, ввиду некоторых изысков планировки города, равна 1 километру.

К перекрёстку может подходить любое количество дорог. Дорог, ведущих от перекрестка к этому же перекрестку, нет. Между любой парой перекрёстков проходит не более одной дороги. Машина дяди Бори может ехать со скоростью v_b км/ч. а k ($0 \leq k \leq 100$) милицейских машин могут ехать со скоростями $v_1 \dots v_k$ км/ч.

Дом Пети расположен у перекрестка 1, а любимая песочница - у перекрестка n . В начальный момент времени милицейские машины находятся на перекрёстках с номерами $r_1 \dots r_k$.

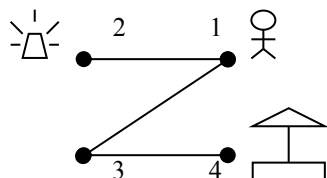
Формат ввода:

n m k количество перекрестков, дорог, и милицейских машин
 v_b v_1 ... v_k скорость машины дяди Бори и k милицейских машин, целые, $\leq 2^{12}$
 p_{11} p_{12}
...
 p_{m1} p_{m2}
 r_1
 t строк - номера перекрестков, соединенных дорогами

r_2 k чисел - номера перекрестков, в которых стоят милицейские машины в начальный момент
 ...
 r_k

Формат вывода: q : -1, если Петя доедет до своей песочницы, и его не поймают, или число от 1 до n - перекресток, в котором поездка Пети закончится.

Комментарий: до песочницы милицейская машина доедет за 45 секунд, а Петя – за 60. И его там будут поджидать.



steal.in

```
4 3 1
120 239
1 3
3 4
2 1
2
```

steal.out

```
4
```

Конденсаторная батарея

Областная олимпиада по информатике
 Гродно, 2001 год

В галактической системе UYUY37UY на планету UU-2765 совершил вынужденную посадку звездолёт "OLIMPIA-2001", у которого вышла из строя конденсаторная батарея, представляющая собой набор соединённых (параллельно и последовательно) конденсаторов. Инженеру-механику известна схема соединения и ёмкости соединённых в батарею конденсаторов. Составьте программу, находящую значение ёмкости батареи.

Схема соединений батареи произвольная, но соединение конденсаторов в батареях только параллельное или последовательное (т.е. соединений типа "треугольник" или "звезда" нет).

При параллельном соединении конденсаторов их общая ёмкость равна сумме соединённых ёмкостей, при последовательном соединении - общая ёмкость вычисляется по формуле: $C_{общ} = C_1 * C_2 / (C_1 + C_2)$

Ёмкость конденсатора - положительное число, не превосходящее 2500, количество конденсаторов в батарее - натуральное число, не превосходящее 50, ёмкость батареи измеряется между спайками с максимальным номером и номером 1.

Формат ввода: первая строка: два натуральных числа n (число конденсаторов в старой батарее) и m (количество спаек конденсаторов), в каждой из следующих n строк по три числа - сведения о конденсаторе: a и b - номера спаек, к которым присоединён конденсатор, c - его ёмкость.

Формат вывода: Одно число - округлённое до трёх знаков после запятой значение ёмкости конденсаторной батареи.

condense.in

```
5 4
1 2 400
1 3 200
2 3 400
3 4 240
3 4 360
```

condense.out

```
240.000
```

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2009 \ День 2 \ 3 - "Странная сеть"*

*codeforces.ru \ Тренировки
2012-2013 Всероссийская командная олимпиада школьников по программированию,
интернет-тур + отборы регионов (ВКОШП 12, Интернет-тур) \ С. Укрепление мостов*

*codeforces.ru \ Архив
1037E. Поездка*

55. Алгоритм Дейкстры. Восстановление пути в алгоритме Дейкстры

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. ком. младшие \ 2003 \ 7 - "Игра "*

Путешествие без остановки

*ACM North Central Regional Programming Contest
1995-96*

Дэвид ненавидит ждать на сигналах светофора во время вождения. Чтобы уменьшить до минимума своё раздражение он подготовил карты различных регионов, где он часто ездит, и измерил среднюю задержку (в секундах) на каждом из всех перекрёстков в этих регионах. Он хочет найти маршруты между заданными точками в этих регионах, которые минимизируют его задержку на перекрёстках (независимо от общего расстояния, которое он должен проехать, чтобы избежать задержек), и обращается за вашей помощью в этом деле.

Формат ввода: Для каждого региона Дэвид предоставляет вам карту. Данные карты идентифицируют сначала некоторое количество перекрёстков n_i . Регионы никогда не включают больше 100 перекрёстков. Перекрёстки в каждом регионе нумеруются последовательно, начиная с номера один (1).

Для каждого перекрёстка, в свою очередь, входные данные определяют затем количество улиц, ведущих от перекрёстка, и для каждой такой улицы номер перекрёстка, к которому ведёт эта улица, и среднюю задержку в секундах, с которой сталкивается Дэвид на этом перекрёстке.

За данными о последнем перекрёстке в регионе появятся номера, ассоциированные с перекрёстками, где Дэвид хочет начать и закончить своё вождение. Все входные данные состоят из последовательности карт со следующим за ними целым нулём (0).

Формат вывода: Для каждого региона, по порядку, распечатайте одну строку вывода, которая содержит номер региона (они также последовательно нумеруются, начиная с 1), список номеров перекрёстков, которые Дэвид встретит на маршруте с минимальной средней задержкой, и среднее количество секунд, которое на которое он будет задержан во время путешествия по этому маршруту. Формат вывода показан в примере ниже.

Маршрут с минимальной средней задержкой всегда будет единственным в каждом регионе. Улица от перекрёстка i до перекрёстка j односторонняя. Для того, чтобы представить двухстороннюю улицу от i до j , карта должна также включать маршрут от перекрёстка j до перекрёстка i . Между перекрёстками i и j не будет больше одного прямого маршрута.

Предположим, Дэвид хочет проехать от перекрёстка 2 до перекрёстка 4 в регионе, показанном на следующей карте:

	От	До	Задержка
1-----2----->3----->4-----5	1	3	3
	1	4	6
	2	1	2
	2	3	7
	2	5	6
+-----+	3	4	5
	5	4	7

Ввод и вывод для этого примера показаны в первом наборе в примере ввода и примере вывода.

nonstop.in

```

5
2 3 3 4 6
3 1 2 3 7 5 6
1 4 5
0
1 4 7
2 4

2
1 2 5
1 1 6
1 2

7
4 2 5 3 13
4 8 5 18
2 3 7 6 14
1 6 6
2 3 5 5 9
3 6 2 7 9
4 6
1 7 2

```

nonstop.out

```

Case 1: Path = 2 1 4; 8 second delay
Case 2: Path = 1 2; 5 second delay
Case 3: Path = 1 2 3 6 7; 20 second delay

```

0
1 7
0

Сумасшедшие гонки

Областная олимпиада по информатике

I тур

Гомель, 9 января 2002 г.

Секретное гомельское общество сумасшедших гонок проводит новый чемпионат по сумасшедшим гонкам. Восхищенные новыми компьютерными автосимуляторами организаторы решили провести соревнование в центре города. Центр города представляет собой трассу для гонок. По ходу проведения сумасшедших гонок у участников стали появляться проблемы. Работники ГАИ, заметив происходящий беспредел, постепенно стали блокировать перекрестки, тем самым мешая гонке.

К счастью у каждого гонщика есть мобильная связь с центром проведения сумасшедших гонок и портативный компьютер. Каждый гонщик знает в определенный момент времени перекресток к которому он едет (или на котором находится), и свою цель. По мобильной связи последовательно поступает информация о перекрытых перекрестках.

Для того, чтобы избежать столкновения с ГАИ, гонщикам необходимо как можно быстрее рассчитать кратчайший путь к заданной цели. Пусть ему в этом поможет ваша программа.

Формат ввода:

n	количество перекрестков в городе ($3 \leq n \leq 100$)
$a_{11} d_{11} a_{12} d_{12} \dots$	перекрестки, с которыми соединён 1-й перекресток
$a_{21} d_{21} a_{22} d_{22} \dots$	перекрестки с которыми соединён 2-й перекресток
\dots	d_{ij} - расстояние между i -м и a_{ij} -м перекрестками ($0 < d_{ij} \leq 100$)
$a_{n1} d_{n1} a_{n2} d_{n2} \dots$	перекрестки, с которыми соединён n -ый перекресток
пустая строка	

Далее следуют запросы, то есть перекрестки, между которыми следует переместиться гонщику (строка "ARRIVE X Y", где X - перекресток, из которого надо доехать до перекрестка Y) и блокируемые перекрестки (строка "LOCK X", где X - блокируемый перекресток). Гарантируется, что изначально каждый перекресток будет иметь хотя бы один соседний, на который можно проехать.

Входной файл заканчивается строкой "EOF" (с переводом на новую строку).

Формат вывода: ваша программа по мере поступления входных данных должна выводить в выходной файл следующие данные:

```
Inquiry n Available! k miles
Road: x1 x2 ... xp
```

Если возможно достигнуть перекрестка y . n - номер запроса, k - минимальное расстояние до перекрестка y , $x_1 x_2 \dots x_p$ - путь от x к y , где $x_1 = x$, а $x_p = y$.

```
Inquiry n Impossible
```

Если невозможно достигнуть y . n - номер запроса.

crazy.in

```

7
2 1 4 1
1 1 3 1
2 1 4 1
1 1 3 1 5 1 6 1
4 1
4 1 7 1
6 1

```

```

ARRIVE 2 7
LOCK 4
ARRIVE 2 7
EOF

```

crazy.out

```

Inquiry 1 Available! 4 miles
Road: 2 1 4 6 7
Inquiry 2 Impossible

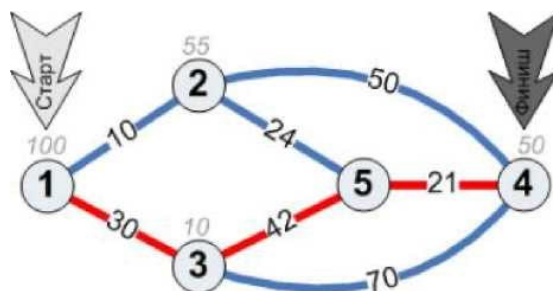
```

Куфический дирхем

Республиканская олимпиада по информатике, II тур
Минск, 2 апреля 2009 г.

Петр Васильевич Колошин всегда был коммуникабельным человеком и, соответственно, имел большое количество друзей. Как-то вечером, просматривая свою электронную почту, Петр Васильевич был приятно удивлен приглашением на юбилей своего студенческого друга Андрея Федоровича и сразу же принял положительное решение. Выяснив, что Андрей Федорович уже много лет увлекается коллекционированием старинных монет и охотится за средневековой серебряной монетой с названием "Куфический дирхем", Петр Васильевич решил непременно подарить эту монету.

Воспользовавшись Интернетом, Петр Васильевич сумел узнать все K городов, где можно приобрести данную монету, а также ее стоимость в каждом из этих городов. Страна, в которой живут Петр Васильевич и Андрей Федорович, насчитывает N городов и M двусторонних автомобильных дорог, каждая из которых связывает два различных города государства. Известно, что Петр Васильевич живет в городе A , а Андрей Федорович - в городе B . Для каждой дороги Петр Васильевич вычислил стоимость проезда с учетом технических характеристик своего автомобиля. В целях экономии Петр Васильевич решил купить монету по пути из города A в город B . Другими словами, маршрут движения Петра Васильевича должен проходить через город, в котором он решит купить монету. Однако оказалось, что ехать через город, в котором монета стоит меньше всего, не всегда выгодно, так как выиграв в стоимости монеты, можно потерять гораздо больше в стоимости дороги и наоборот: Ваша задача - помочь Петру Васильевичу выбрать оптимальный маршрут и город Z , где следует приобрести монету. Маршрут должен начинаться в городе A , заканчиваться в городе B и проходить через город Z . Стоимость данного маршрута должна быть минимальной. Под стоимостью маршрута будем понимать сумму количества денег, потраченных на дорогу и стоимость монеты в городе Z . Ниже приведен пример для $N = 5$, $M = 7$, $A = 1$, $B = 4$.



Для данного примера $K = 4$, стоимость монеты обозначена сверху над кругом, обозначающим город. Оптимальный маршрут выделен красным цветом. Для данного примера $Z = 3$.

Формат ввода:

Первая строка входного файла содержит три целых числа N , M и K ($2 < N < 2000$; $1 < M < 100000$; $1 \leq K < N$), где N - количество городов в стране, M - количество дорог, а K - количество городов, в которых продается искомая монета. Будем считать, что все города пронумерованы целыми числами от 1 до N .

Вторая строка входного файла содержит два целых числа A и B ($1 < A, B < N$; $A \neq B$), где A - номер города, в котором живет Петр Васильевич, а B - номер города, в котором живет Андрей Федорович.

Третья строка содержит K пар целых чисел V_i и C_i ($1 < V_i < N$; $1 < C_i < 10^9$), где V_i - это номер города, в котором можно приобрести искомую монету, а C_i - стоимость монеты в соответствующем городе. Известно, что $V_i \neq V_j$, если $i \neq j$. Все числа в строке разделены одиночными пробелами.

Каждая последующая из M строк содержит три числа X_i , Y_i , S_i ($1 < X_i, Y_i < N$; $X_i \neq Y_i$; $1 < S_i < 10^5$), где X_i и Y_i - номера городов, связанных двусторонней дорогой, а S_i - стоимость проезда по данной дороге. Не существует двух различных дорог, связывающих одни и те же города.

Формат вывода:

Единственная строка выходного файла должна содержать одно целое число - минимальную стоимость маршрута. Гарантируется, что решение существует.

Пример 1

input.txt

```
3 3 2
3 1
1 20 2 5
1 2 7
1 3 5
2 3 8
```

output.txt

```
20
```

Маршрут, город Z

```
{3, 2, 1}
Z = 2
```

Пример 2

input.txt

```
5 7 4
1 4
1 100 4 50 3 10 2 55
1 2 10
5 3 42
```

output.txt

```
103
```

Маршрут, город Z

```
{1, 3, 5, 4}
Z = 3
```

```

1 3 30
2 4 50
3 4 70
2 5 24
4 5 21

```

Пример 3**input.txt**

```

8 7 1
1 6
5 187
1 8 32
8 6 39
5 4 51
1 4 101
2 4 17
3 7 46
2 8 23

```

output.txt

440

Маршрут, город Z

```

{1, 8, 2, 4, 5, 4, 2, 8, 6}
Z = 5

```

dl.gsu.by \ Олимпиады по информатике
USACO \ 2010 \ December \ Silver \ 1 - "Apple Delivery"

dl.gsu.by \ Олимпиады по информатике
USACO \ 2011 \ March \ Silver \ 2 - "Package Delivery"

dl.gsu.by \ Олимпиады по информатике
USACO \ 2012 \ February \ Silver \ 3 - "RELOCATE"

Комментарий: $M \leq 500\,000$, $0 \leq L \leq 1\,000$

56. Эйлерова цепь. Эйлеров цикл. Эйлеров граф. Построение эйлерова цикла

В городе Hot-Dog

Автор: Игорь Дорошев

В городе Hot-Dog City живёт мальчик Веня. И он очень любит поесть. Больше всего ему нравятся чебуреки.

В городе M чебуречных и N перекрёстков. У каждой чебуречной есть вход с одного перекрёстка и выход на другой перекрёсток. Мальчик Веня хочет обойти все чебуречные, и при этом ему нельзя посещать одну и ту же чебуречную дважды, так как за один раз он уничтожает все чебуреки.

Через чебуречную можно идти только в одном направлении. Перекрёсток, в котором Веня начинает маршрут, не обязательно соответствует перекрёстку, в котором он заканчивает.

Помогите Вене выбрать маршрут. Решение всегда существует.

Формат ввода: в первой строке входного файла находятся числа N и M ($0 < N, M < 100$). Далее следуют M строк, содержащих числа a и b ($0 < a, b < N+1$) - перекрёсток входа и перекрёсток выхода для i -той чебуречной соответственно.

Формат вывода: количество и номера перекрёстков в порядке обхода, которые Веня должен посетить.

Пример 1

input.txt

```
4 5
1 2
2 3
2 3
3 2
3 4
```

output.txt

```
6
1 2 3 2 3 4
```

Пример 2

input.txt

```
3 3
1 2
2 3
3 1
```

output.txt

```
4
1 2 3 1
```

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2010 \ Областная для 1-8 кл (засвечена 26 марта) \ 7 - 8 кл \ 8 - "Граф"

codeforces.ru \ Тренировки

2015, Отборочный контекст СГАУ на четвертьфинал ACM ICPC \ С. История Принцессы

57. Список рёбер. Двусвязные списки. Элементы дерева. Глубина узла, время входа и выхода

codeforces.ru \ Тренировки

2012-2013, Отборочный контекст СГАУ на четвертьфинал ACM ICPC \ J. Последнее слово техники

dl.gsu.by \ Олимпиады по информатике

Гомельская обл. \ 2011 \ День 1 \ 3 - "Водородный поезд"

dl.gsu.by \ Олимпиады по информатике

Гомельская обл. \ 2013 \ День 1 \ 4 - "Замечательная карта"

[codeforces.ru \ Тренировки](#)
[2012-2013 Тренировка СПбГУ В #2 Поиск в глубину \ А. Предок](#)

[codeforces.ru \ Тренировки](#)
[2011-2012 Командный чемпионат школьников Санкт-Петербурга по программированию \(XIX\) \ Г. Шаги](#)

[codeforces.ru \ Тренировки](#)
[2011-2012 Всероссийская командная олимпиада школьников по программированию \(ВКОШП 11\) \ К. Королевская династия](#)

58. Минимальное остовное дерево. Система непересекающихся множеств. Алгоритм Крускала

Телефонная сеть

В некоторой местности расположены несколько городов, известны их координаты (x,y) . Телефонная компания собирается объединить эти города в единую телефонную сеть таким образом, чтобы от каждого из городов можно было дозвониться до каждого из оставшихся напрямую или хотя бы через несколько промежуточных городов. Цель компании – построить сеть, самую экономичную по длине прокладываемого кабеля.

Напишите программу, которая определит минимальную длину кабеля, необходимую для объединения всех городов в сеть.

Формат ввода: первая строка – количество городов, натуральное число, ≤ 100 . Остальные строки – координаты городов. $-2\,000\,000\,000 \leq x,y \leq 2\,000\,000\,000$.

Формат вывода: искомая минимальная длина кабеля с точностью два знака после точки.

input.txt

```
5
0 0
-4 0
1 0
1 2
4 0
```

output.txt

```
10.00
```

Порядок городов

Автор: Алексей Ропан

Новый русский купил огромный участок земли с несколькими городами. К сожалению, дороги между этими городами испорчены.

Новый русский решил восстановить дороги между городами, он хочет сэкономить и, поэтому, будет строить сеть дорог минимальной длины так, чтобы все города были включены в эту сеть. Помогите ему построить такую сеть.

Формат ввода:

n количество городов ($0 < n < 21$)
 $x_1 y_1 s_1$ $x_i y_i$ - координаты i -го города ($-10 \leq x_i, y_i \leq 10$)
 $x_2 y_2 s_2$ s_i -название i -го города ($0 < \text{длина } s_i < 11$)
 ... координаты и название города разделены одним
 $x_n y_n s_n$ или несколькими пробелами

Формат вывода: определите порядок городов по возрастанию и столицу. Меньшим будет тот город, который имеет меньшее количество дорог, а если количество дорог одинаковое, тот, чье название раньше идёт по алфавиту.

Столицей является город с наибольшим количеством дорог, а если количество дорог одинаковое, тот, чье название раньше идёт по алфавиту.

Выведите города в порядке возрастания, а затем с новой строки "Capital: " и название столицы.

cities.in

```
6
1 1 Gomel
-1 1 Brest
3 1 Minsk
3 -1 Grodno
3 2 Vitebsk
4 1 Mogilev
```

cities.out

```
Brest Grodno Mogilev Vitebsk Gomel Minsk
Capital: Minsk
```

Строительство дорог

В некоторой местности есть k ($0 < k < 100$) населённых пунктов. Между некоторыми из них проложены асфальтированные дороги. Идя навстречу пожеланиям жителей, муниципальные службы района решили проложить дополнительные дороги так, чтобы из любого пункта в любой другой можно было проехать по асфальту напрямую или проезжая через другие пункты.

Посчитайте сколько дорог n по минимуму надо построить и определите минимальную общую длину дорожного полотна l , которое придётся проложить. Расстояния между всеми населёнными пунктами известны.

Формат ввода:

k
 $s_{1,2}$
 ...
 $s_{1,k}$
 $s_{2,3}$
 ...
 $s_{2,k}$
 ...
 $s_{k-1,k}$
 где $s_{i,j}$ – расстояние между i и j пунктами ($-120 \leq s_{i,j} \leq 120$ и $s_{i,j}$ не равно 0).

Все расстояния – целые числа. Знак минус в расстоянии означает, что между этими пунктами нет асфальтированной дороги.

Формат вывода:

<i>n</i>	
<i>l</i>	
input.txt	output.txt
5	2
-10	18
-15	
-17	
-22	
7	
-8	
-13	
-13	
-9	
26	

dl.gsu.by \ Подготовка к IOI 2005

Структуры данных \ Стандартные задачи \ 2 - "Система непересекающихся множеств"

dl.gsu.by \ Методы алгоритмизации

Задачи для книги \ Графы \ Контрольные задачи \ Г12 - "МИНИМАЛЬНОЕ ОСТОВНОЕ ДЕРЕВО"

dl.gsu.by \ Методы алгоритмизации

Олимпиады по информатике \ Сложные структуры данных \ Система непересекающихся множеств (Лес) \ 08_Gust - "Острова"

dl.gsu.by \ Олимпиады по информатике

Белорусская \ 2014 \ День 2 \ 3 - "Дороги"

dl.gsu.by \ Олимпиады по информатике

USACO \ 2008 \ November \ Gold \ 2 - "Cheering up the Cows"

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2013 \ Городская 1-11 кл, 9 ноября \ 9 - 11 кл \ 12 - "Ножницы"

[codeforces.ru \ Архив](#)

[939D. Спасение любви](#)

[codeforces.ru \ Архив](#)

[1095F. Сделай связным](#)

codechef.com \ Practice(easy)
Houses and Restaurants

codeforces.ru \ Тренировки
2017-2018 Цикл интернет-олимпиад. Третья командная олимпиада, продвинутый уровень
(11 ноября 2017 года) \ F. Тренировки Тора

codeforces.ru \ Тренировки
Тренировочный контекст МПГУ 2018-2019 \ F. Айландлэнд

codeforces.ru \ Архив
1245D. Шичикуджи и электросеть

59. Наименьший общий предок

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 9 - "Ближайший поворот"

acm.timus.ru
1471. Расстояние в дереве

dl.gsu.by \ Олимпиады по информатике
Казахстанская \ 2011 \ Day_2 \ F - "Выборы"
Комментарий: в стандартном тесте третья строка файла ввода должна содержать "2 3"
Комментарий: полагайте, что для решения задачи достаточно алгоритма линейного подъёма

dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2018 \ День 1 \ 4 - "Дерево удачи"

codeforces.ru \ Архив
1051F. Самое короткое условие

60. Префиксное дерево

dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Городская 1-9 кл, 19 апреля \ 8 - 9 кл \ 12 - "Много слов"

*dl.gsu.by \ Олимпиады по информатике
Российская (ком.,Inet) \ 2006 \ Базовые \ Олимпиада_3 (21 октября 2006) \ 8 - "Различные слова"*

*dl.gsu.by \ Олимпиады по информатике
Канадская (Stage 1) \ 2003 \ 6 - "S4_SUBSTR"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2019 \ День 2 \ 3 - "Текстовый редактор"*

*codeforces.ru \ Тренировки
2014-2015 Цикл интернет-олимпиад. Бонусная командная олимпиада (16 мая 2015 года) \ I. Tree*

*codeforces.ru \ Архив
817E. Выбор командира*

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2019 \ День 1 \ 4 - "Инопланетные захватчики"*

61. Дерево отрезков

*dl.gsu.by \ Олимпиады по информатике
Севастопольская школа \ 2012 \ Day 2 \ К - "Дима и массив"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская гор. \ 2013 \ Областная 1-9 кл, 3 мая \ 8 - 9 кл \ 12 - "Новое устройство"*

*dl.gsu.by \ Олимпиады по информатике
Севастопольская школа \ 2012 \ Day 2 \ В - "Range Variation Query"*

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2015 \ День 2 \ 2 - "Скобковедение"*

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2009 \ День 1 \ 3 - "Адаптивный поиск"*

e_j

2017-08-13 [Дорешивание] Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ I. Саша и строки

[codeforces.ru \ Группы \ КЮП](https://codeforces.ru/group/KYOP)
Сборы #3 \ С. Просто сумма

[codeforces.ru \ Группы \ КЮП](https://codeforces.ru/group/KYOP)
Сборы #3 \ Е. USACO style

[codeforces.ru \ Группы \ КЮП](https://codeforces.ru/group/KYOP)
Сборы #3 \ F. Не USACO style

Космодром «Центральный»

Областная олимпиада по информатике

Гомель, 2010 год

Соавтор тестов: Илья Смоляр

2148 год. Более века назад люди освоили практически все планеты нашей Вселенной, и межгалактические путешествия стали обыденностью. Космодром «Центральный» специализируется на запуске космических грузовых кораблей.

Пусковые двигатели всех космических кораблей работают на солнечной энергии. В зависимости от грузоподъемности кораблю для запуска может требоваться различное число эрстедов. Эрстед – единица измерения солнечной активности.

Каждый рабочий день космодрома начинается с того, что отдел прогнозирования космодрома предоставляет отделу работы с клиентами прогноз солнечной активности на ближайшие N дней начиная с завтрашнего. Прогноз можно представить в виде последовательности из N целых чисел A_j , где A_j - прогнозируемое число эрстед солнечной активности в j -й день.

Отдел работы с клиентами занимается приёмом и обработкой заявок на запуск. Каждая i -я заявка описывается при помощи трех целых чисел S_i , F_i и R_i . S_i - минимальное количество дней, которое должно пройти между запуском корабля и моментом подачи заявки, F_i - соответственно максимальное количество дней и R_i - требуемое для запуска количество эрстед. Результатом обработки i -й заявки является число $|A_{L_i} - R_i|$ минимальное из возможных.

Космодром действительно большой и способен запускать в день практически неограниченное количество кораблей. Ваша задача - разработать программу, которая по имеющемуся прогнозу позволит обработать все принятые за день заявки.

Формат ввода: первая строка входного файла содержит два целых числа N и M , разделённых одиночным пробелом ($1 \leq N, M \leq 65\,536$). N - количество дней в прогнозе, M - количество поданных за день заявок.

Вторая строка входного файла содержит N целых чисел A_j ($1 \leq A_j \leq 10^9$). Числа в строке разделяются одиночными пробелами.

Каждая из следующих M строк описывает одну заявку и содержит три целых числа S_i , F_i и R_i ($1 \leq S_i \leq F_i \leq N$; $1 \leq R_i \leq 10^9$). Числа в строке разделяются одиночными пробелами.

Формат вывода: выходной файл должен содержать M целых неотрицательных чисел L_i , где L_i - результат обработки i -й заявки. Порядок следования чисел L_i в выходном файле должен соответствовать порядку ввода заявок. Если для какой-либо заявки несколько вариантов ответа, то достаточно вывести любой.

input.txt

```
5 2
1 3 2 9 5
1 3 3
2 4 7
```

input.txt

```
10 3
81 70 52 19 85 41 78 62 98 31
4 9 81
1 7 30
3 8 52
```

output.txt

```
0
2
```

output.txt

```
3
11
0
```

dl.gsu.by \ Олимпиады по информатике
Казахстанская \ 2015 \ Day_1 \ A - "Контейнеры и отсеки"

codeforces.ru \ Группы \ КЮП
Сборы #3 \ В. Муравьиная колония

codeforces.ru \ Группы \ КЮП
Сборы #3 \ Г. Олимпиада по информатике

codeforces.ru \ Тренировки
2012-2013 Тренировка СПбГУ В #1 \ J. Range Minimum Query

codeforces.ru \ Архив
339D. Ксюша и битовые операции

codeforces.ru \ Архив
831E. Сортировка карт

codeforces.ru \ Архив
558E. Простое задание

[codeforces.ru \ Архив](#)
[689D. Друзья и последовательности](#)

[codeforces.ru \ Архив](#)
[381E. Сережа и скобочки](#)

[codeforces.ru \ Архив](#)
[899F. Удаление букв](#)

[codeforces.ru \ Архив](#)
[609F. Лягушки и комары](#)

[codeforces.ru \ Архив](#)
[811E. Владик и интересные флаги](#)

62. N-мерные рекуррентные соотношения

[codeforces.ru \ Тренировки](#)
[2015-2016 ACM-ICPC Southwestern Europe Regional Contest \(SWERC 15\) \ E. Wooden Signs](#)

[codeforces.ru \ Тренировки](#)
[2000-2001 Всероссийская командная олимпиада школьников по программированию \(ВКОШП 00\) \ E. Симпатичные узоры](#)

Землетрясение

Районная олимпиада по информатике

Мозырь, 2001 год

Соавтор тестов: Владислав Вишневский

В крупном городе произошло сильное землетрясение. Одно из высотных зданий оказалось непрочным, и его конструкция начала разрушаться. Некоторые комнаты оказались частично заваленными обломками, и некоторые стены разрушились.

В момент землетрясения Билл сидел в своем офисе компании "BigSoft", расположенном в этом здании на k -ом этаже. Ему очень повезло, что он выжил. Теперь ему предстоит как можно быстрее выбраться из здания. Но главная трудность в том, что все лестницы в здании оказались разрушенными, а лифты - уничтоженными. Не растерявшись и вооружившись портативным отбойным молотком известной фирмы "MicroHard", он решил "прорываться" к первому этажу. К счастью, у него оказался полный план того, что осталось от здания после катастрофы.

Проблема Билли состоит в том, что он не умеет программировать, и он надеется на то, что Вы поможете ему. Вы должны рассчитать минимальное время на то, чтобы добраться к одному из

выходов. Все выходы расположены на первом этаже. Выход - это отсутствие стены на любом из участков, расположенных по периметру первого этажа. Гарантировано существование по крайней мере одного пути к выходу. Пробираться он может только с верхнего этажа на нижний этаж при помощи "бурения" перегородки, разделяющей этажи (т.е. пола относительно верхнего этажа). Время, потраченное на выход из здания, будет равно сумме времён, потраченных на бурение каждого отверстия. Временем на перемещения по этажу можно пренебречь. Естественно, чем больше завален участок, тем больше времени Биллу понадобится на работу. Делать отверстия в стенах нельзя.

Каждый этаж представляет собой прямоугольник, разбитый на $m \times n$ участков (m строк, n столбцов). В момент трагедии Билл находился на участке $[x,y]$ (x - строка, y - столбец) этажа номер k , во входном файле задается ровно k этажей. Каждый участок каждого этажа задается в виде $d[i,j]$ $c[i,j]$, где:

$d[i,j]$ - подмножество символов {N,S,W,E}. (N - север, на плане этажа это соответствует верхней стене, S - юг, соответствует нижней стене, W - запад, соответствует левой стене, E - восток, соответствует правой стене). Нахождение соответствующего символа в $d[i,j]$ означает, что с этой стороны у участка находится стенка. Если вокруг участка нет стен, то $d[i,j]=\{\}$. Данные о соседних участках не противоречат друг другу.

$c[i,j]$ - время бурения пола в этом участке (целое, $0 \leq c[i,j] \leq 50000$). Время бурения на первом этаже никак не влияет на решение (пол на первом этаже бурить нельзя).

Формат ввода:

```
m n k
x y
d[1,1] c[1,1]
d[1,2] c[1,2]
...
d[m,n] c[m,n]
...
d[1,1] c[1,1]
d[1,2] c[1,2]
...
d[m,n] c[m,n]
```

$0 < n, m, k \leq 125$, $n * m * k \leq 500000$, $0 < x \leq m$, $0 < y \leq n$. Все числа целые. d - этажи здания (начиная с k -го и до первого): $n * m$ строк, описывающих каждый участок k -го этажа, далее описания этажей $k-1$, $k-2$, ..., и наконец $n * m$ строк, описывающих каждый участок первого этажа. Описания этажей разделяются пустой строкой.

Формат вывода: общее время, необходимое для выхода из здания.

Комментарий к примеру ввода: этаж имеет размер 2×2 , всего 2 этажа. Рабочее место Билли (2, 2).

Этаж 2:

1	1
4	5

Этаж 1:

1	2
3	4

Комментарий к примеру вывода: возможная последовательность перемещений:

Этаж 2:

(2, 2) -> (2, 1) -> (бурим)

Этаж 1:

(2, 1) -> (1, 1) -> (выход)

shake.in

2 2 2

2 2

{W, N, S} 1

{E, S, N} 1

{W, N, S} 4

{E, N, S} 5

{W, E} 1

{N, W, E} 2

{W, S, E} 3

{W, E} 4

shake.out

4

Бит на бит не приходится

Областная олимпиада школьников по информатике

Гомель, 10 января 2003г.

Задана отсортированная последовательность всех N-битных чисел, в которой были удалены все числа, имеющие в своей двоичной записи более, чем L единиц. Вам необходимо найти I - тый элемент этой последовательности. Нумерация элементов начинается с единицы.

Формат ввода:

N L I

$1 \leq N \leq 31$. $1 \leq L \leq N$

Формат вывода: I-й элемент последовательности

bit.in

3 1 3

bit.out

010

informatics.mccme.ru

Задача №112750. Укладка плитки

informatics.mccme.ru

Задача №112820. Подводная лодка

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2015 \ Школьная 1-11 кл, 7 октября \ 9 - 11 кл \ 8 - "Палиндром"

*dl.gsu.by \ Олимпиады по информатике
USACO \ 2006 \ November \ Silver \ 3 - "Round Numbers"*

*dl.gsu.by \ Олимпиады по информатике
COCI \ 2014 \ coci_2013-14 \ 14_COCI5_Feb \ 4 - "DOMINE"*

*dl.gsu.by \ Олимпиады по информатике
Белорусская \ 2014 \ День 1 \ 3 - "Потребитель"*

63. Персистентное дерево отрезков

*codeforces.ru \ Группы \ КЮП
Персистентное дерево отрезков \ А. Скука*

*codeforces.ru \ Тренировки
2017 Открытый Чемпионат ЯрГУ по спортивному программированию \ J. Уникальные суммы*

*codeforces.ru \ Группы \ КЮП
Персистентное дерево отрезков \ В. Создание армии*

*codeforces.ru \ Группы \ КЮП
Персистентное дерево отрезков \ С. Пока я срываюсь*

*codeforces.ru \ Архив
1080F. Катя и наборы отрезков*

64. Признак ромба. Обход вершин четырёхугольника. Признак квадрата. Каноническое и общее уравнения прямой. Параллельность прямых. Совпадение прямых

*dl.gsu.by \ Олимпиады по информатике
Гом. обл. младшие \ 2003 \ День 2 \ 4 - "Фермер "*

Ромб

Автор: Алексей Ропан

Определите, являются ли заданные точки вершинами ромба.

Формат ввода: четыре строки с координатами точек.

Формат вывода: "да" или "нет".

input.txt

```
0 0
0 3
-1 1.5
1 1.5
```

output.txt

```
да
```

Marsianius

Автор: Александр Кулицкий

Вот и до Марса добралась цивилизация! Наконец-таки марсиане договорились с землянами о поставке на Марс компьютеров.

После того, как первая партия уже попала на Марс, обнаружили, что земные мыши очень неудобны для квадратных рук марсиан и поэтому земляне теперь вынуждены производить специальные мыши для марсиан. Поручено это дело фирме Genius, а точнее их новому подразделению - Marsianius.

Земляне очень уважают марсиан и поэтому не хотят, чтобы на Марс попали бракованные, не квадратные мыши. Вам поручили проверять мыши перед отправкой.

Формат ввода: четыре строки с координатами точек.

Формат вывода: "да", если мышь квадратная, и "нет" - в обратном случае.

mouse.in

```
0 0
1.5 1.5
1.5 0
0 1.5
```

mouse.out

```
да
```

Параллельные отрезки

Автор: Егор Казаченко

Определите, параллельны ли отрезки.

Формат ввода: первые две строки – координаты концов первого отрезка, следующие две строки – координаты концов второго отрезка.

Формат вывода: "да" или "нет".

input.txt

```
1 1
3.5 3.5
1 -1
2 -2
```

output.txt

```
нет
```

Задание по геометрии

Автор: Дмитрий Барбашов

Вася Пупкин, вернувшись домой, обнаружил, что ему по геометрии задано 10 задач на определение принадлежности точки к прямой. Но велика Пупкинская лень и он решил, долго не думая, обратиться к вам за помощью.

Формат ввода: первая строка - координаты испытываемой точки. Следующие две строки - координаты двух точек, определяющих прямую.

Формат вывода: "принадлежит" или "не принадлежит".

input.txt

2 0
0 0
3 0

output.txt

принадлежит

Задание по геометрии 2

Автор: Дмитрий Барбашов

Воодушевленный успехом своей домашней работы, Вася Пупкин и на следующий день решил обратиться к вам. Теперь ваша задача - определить характер взаимного расположения двух прямых.

Формат ввода: первые две строки – координаты точек, принадлежащих первой прямой. Следующие две строки – координаты точек, принадлежащих второй прямой.

Формат вывода: "пересекаются", или "параллельны", или "совпадают".

input.txt

0 0
9 0
1 0
3 0

output.txt

совпадают

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2014 \ Школьная 1-11 кл, 10 октября \ 9 - 11 кл \ 10 - "Точки на прямой"

Прямые и точки

Авторы: Алексей Ропан, Андрей Левковский

Определите характер взаимного расположения точек и прямых.

Формат ввода: строки одного из следующих видов:

$a \ b \ c$ коэффициенты очередной прямой, $-60\ 000 \leq a, b, c \leq 60\ 000$
 $x \ y$ координаты точки, $-60\ 000 \leq x, y \leq 60\ 000$

За каждым описанием прямой следуют ноль или больше описаний точек.

Формат вывода: для каждого запроса (описания прямой и описаний точек, следующих за

описанием прямой) выведите информацию в формате, указанном в примере вывода.

input.txt

```
1 0 -1
1 1
2 2
2 3
2 -3 2
1 1
2 2
2 3
2 -2 2
2 -2 2
1 1
2 2
2 3
```

output.txt

```
Прямая: 1
Точка 1 принадлежит прямой
Точка 2 не принадлежит прямой
Точка 3 не принадлежит прямой
Прямая: 2
Точка 1 не принадлежит прямой
Точка 2 принадлежит прямой
Точка 3 не принадлежит прямой
Прямая: 3
Прямая: 4
Точка 1 не принадлежит прямой
Точка 2 не принадлежит прямой
Точка 3 принадлежит прямой
```

dl.gsu.by \ Программирование – начинающие (P/O)

1-10 \ 3 \ 4 - "Умный математик"

Комментарий: в задаче полагайте, что трапеция – это четырёхугольник, у которого хотя бы две стороны параллельны

65. Вектора. Параллельность векторов. Ноль-вектор. Направляющий и нормальный вектора. Уравнение прямой по точке и направляющему вектору. Построение перпендикулярной прямой

Наборы

Авторы: Алексей Ропан, Андрей Левковский

В файле находится неопределённое количество наборов данных. Каждый набор представляет собой количество векторов и их координаты.

Найдите все параллельные вектора и выведите их в формате, который указан ниже.

Формат ввода:

n количество векторов,
 $x_1 \quad 1 \leq n \leq 100$
 y_1 координаты вектора, -
 $x_2 \quad 100 \leq x_i, y_i \leq 100$
 y_2
 \dots
 x_n
 y_n

Формат вывода: для каждого набора выведите "*Набор #k*", где k - номер набора, и на следующих строках "*i: 1 2 ... n*", где номера $1..n$ - номера векторов, которые параллельны i -му вектору.

input.txt

```
4
0 0
2 12
1 5
1 6
0
3
1 1
1 2
1 3
```

output.txt

```
Набор #1
1: 2 3 4
2: 1 4
3: 1
4: 1 2
Набор #2
Набор #3
1:
2:
3:
```

Параллельность

Автор: Алексей Ропан

Дано некоторое количество векторов и их координаты. Определите наибольшее количество векторов, которые параллельны между собой, и их порядковые номера.

Формат ввода:

n количество векторов ($1 \leq n \leq 100$)

$x_1 y_1$

... $x_i y_i$ - координаты i -ого вектора ($-100 \leq x_i, y_i \leq 100$)

$x_n y_n$

Формат вывода: Выведите наибольшее количество векторов, параллельных между собой в группе (если таких групп несколько, выведите то, у которой первые порядковые номера меньше), и с новой строки их порядковые номера, начиная с наименьшего.

input.txt

```
10
-2 -3
5 4
2 3
4 6
-5 -4
-10 -8
-12 -18
1 5
24 36
5 -1
```

output.txt

```
5
1 3 4 7 9
```

MarsPoint

Авторы: Андрей Левковский, Александр Кулицкий

Вот и готова первая партия квадратных мышек от фирмы Marsianius! Осталось только её

доставить. К сожалению, у землян нет такой возможности, поэтому было решено, что марсиане прилетят на Землю.

Так как марсиане умеют ходить только прямо, то их корабль своим центром должен приземлиться в такой точке, чтобы они смогли дойти до завода по прямой дороге, не встретив на своём пути преград. Дорога задаётся своей фиксированной точкой $M_0(x_0, y_0)$ – завод - и направляющим вектором $\vec{a}(a_1, a_2)$.

Земляне нашли подходящую точку для приземления, но при передаче информации из-за космической бури к марсианам попали искажённые данные, и они не знают, смогут ли попасть на завод. Вам, как главному математику планеты, поручено проверить, находится ли центр марсианского корабля $M(x, y)$ на заданной прямой.

Формат ввода:

$x\ y$
 $x_0\ y_0$
 $a_1\ a_2$

Формат вывода: "да", если марсиане приземлились правильно, и "нет" в ином случае.

point.in

0 0
1 1
2 2

point.out

да

Перпендикулярность

Автор: Сергей Коноплич

Определите, принадлежит ли точка $M(x, y)$ прямой p_2 , которая проходит через точку $M_0(x_0, y_0)$ и перпендикулярна исходной прямой p_1 , заданной своей фиксированной точкой $M_0(x_0, y_0)$ и своим направляющим вектором $a(a_1, a_2)$.

Формат ввода:

$x\ y$
 $x_0\ y_0$
 $a_1\ a_2$

Формат вывода: "да" или "нет".

input.txt

0 0
1 1
2 2

output.txt

нет

Прямые и точки 2

Автор: Алексей Ропан

Прямая задана направляющим вектором и фиксированной точкой. Через эту точку проведён перпендикуляр к исходной прямой.

Определите характер взаимного расположения некоторых точек и перпендикуляра.

Формат ввода: Строки одного из следующих видов:

$x_0 y_0 a_x a_y$ координаты фиксированная точки и направляющего вектора соответственно
 $x y$ координаты очередной испытуемой точки
 -2 147 483 648 < x, y, x_0, y_0, a_x, a_y < 2 147 483 647

За каждым описанием прямой следуют ноль или больше описаний точек, в конце файла ввода один перевод строки.

Формат вывода: Для каждого запроса (описания прямой и описаний точек, следующих за описанием прямой) выведите информацию в формате, указанном в примере вывода.

Комментарий: ограничение по времени – 3 секунды.

input.txt

```
2 2 0 1
0 2
2 3
2 2
0 0 1 -5
0 0 -1 -1
-100 -100
```

output.txt

```
Перпендикуляр: 1
Точка 1 принадлежит
Точка 2 не принадлежит
Точка 3 принадлежит
Перпендикуляр: 2
Перпендикуляр: 3
Точка 1 не принадлежит
```

Мосты

Областная олимпиада школьников по информатике
 Гомель, 11 января 2003 г.

Около одного города есть большое озеро. Было решено построить несколько мостов над озером. Так несколько опор были подняты над водой. Вам даны их координаты. Дальнейшее формирование мостов происходит согласно следующим правилам.

Первый мост формируется на прямой, соответствующей уравнению $Ax + By + C = 0$. Затем - другой мост, перпендикулярный к предыдущему, формируется в каждой точке с опорой, пересечённой первым мостом. Затем формируются следующие перпендикулярные мосты, начинаясь с каждой опоры, пересечённой ранее построенными мостами, и так далее.

Ваша задача - определить окончательное число опор, не пересечённых ни одним из мостов.

Формат ввода:

n количество опор ($1 \leq n \leq 50$)
 $a b c$ целые коэффициенты прямой для первого моста. $-100 \leq a, b, c \leq 100$
 $x_1 y_1$
 ...
 $x_i y_i$ x_i, y_i – целые координаты i -ой опоры. $-21 * 10^{14} \leq x_i, y_i \leq 21 * 10^{14}$
 $x_n y_n$

Формат вывода: s – количество неиспользованных опор.

bridges.in

```
12
1 -1 0
2 2
-2 -2
8 5
```

bridges.out

```
4
```

```
7 3
6 -2
5 0
5 1
5 3
5 4
4 1
4 2
4 4
```

*dl.gsu.by \ Олимпиады по информатике
Гомельская обл. \ 2008 \ День 1 \ 1 - "Оборона Бейтланда"*

66. Расстояние от точки до прямой. Пересечение прямых. Метод Крамера

От точки до прямой

Автор: Антон Надёжкин

Определите расстояние от точки до прямой.

Формат ввода: первые две строки содержат координаты двух точек, через которые проходит прямая, следующая строка – координаты испытываемой точки.

Формат вывода: расстояние с точностью два знака после запятой.

input.txt

```
0 1
1 0
0 0
```

output.txt

```
0.71
```

*dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 8 \ 3 - "Три деревни"*

Комментарий: определяйте расстояние от деревни до прямой, соединяющей две другие деревни

*dl.gsu.by \ Программирование – начинающие (P/O)
11-20 \ 9 \ 3 - "Магистраль"*

Точка пересечения

Автор: Евгений Грицкевич

Определите точку пересечения прямых.

Формат ввода: первые две строки – координаты двух точек, принадлежащих первой прямой, следующие две строки – координаты двух точек, принадлежащих второй прямой.

Формат вывода: координаты точки пересечения с точностью два знака после запятой или "нет", если прямые не пересекаются. Если прямые совпадают, выведите "совпадают".

input.txt

```
1 1
3.5 3.5
1 -1
2 -2
```

output.txt

```
0.00 0.00
```

Периметр

Автор: Алексей Ропан

Дано некоторое количество прямых и их коэффициенты a , b , c . Найдите периметр многоугольника, получившегося в результате пересечения прямых.

Прямые заданы в порядке обхода сторон многоугольника. Многоугольник задан без самопересечений. Две смежные прямые имеют только одну общую точку.

Формат ввода:

n количество прямых ($1 \leq n \leq 10$)

$a_1 b_1 c_1$

$a_2 b_2 c_2$ коэффициенты i -ой прямой

... ($-32768 \leq a_i, b_i \leq 32768$, $-536\,870\,912 \leq c_i \leq 536\,870\,912$)

$a_n b_n c_n$

Формат вывода: выведите длину периметра многоугольника согласно примеру вывода.

lines.in

```
4
-1 0 2
0 -5 5
-9 0 45
0 1 3
```

lines.out

```
14.00
```

67. Расположение точек относительно прямой. Проекция точки на прямую. Принадлежность проекции точки отрезку прямой. Построение симметричной точки

А где точка?

Автор: Егор Казаченко

В один прекрасный день Уасе задали задачу по геометрии. Ему была дана точка (x_1, y_1) и направляющий вектор (a_x, a_y) . Точка и вектор задавали положение прямой. Уаса внимательно прочитал условие, и заметил, что была дана ещё одна точка (x_0, y_0) . Немного позднее он

заставил себя перечитать основную часть условия опять. Там было сказано, что от точки (x_0, y_0) надо опустить перпендикуляр к прямой, а затем продлить перпендикуляр на его длину.

Уаса не может додуматься, где будет точка конца "удлинённого перпендикуляра", если его начало в точке (x_0, y_0) . Помогите Уасе!

Формат ввода: первая строка содержит координаты точки (x_0, y_0) , вторая строка содержит координаты точки (x_1, y_1) , третья строка содержит два числа (a_x, a_y) , характеризующих направляющий вектор.

Формат вывода: ответ на задачу. Два числа, выведенных через пробел, с двумя знаками после запятой.

uasa.in

0 4
0 0
10 10

uasa.out

4.00 0.00

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинающие, ком. \ 2 - "Симметрия"*

Комментарий: вывод в стандартном тесте:

-1.00
-1.00

*dl.gsu.by \ Олимпиады по информатике
GCSW \ 2001 \ Начинающие, ком. \ 3 - "Ориентирование"*

*dl.gsu.by \ Программирование – начинающие (P/O)
1-10 \ 8 \ 3 - "Таксофон"*

Алёша Попович и конь бабы Яги

Автор: Алексей Ропан

Однажды Алёша Попович заблудился в лесу после очередного сражения, в котором он потерял коня. После двухчасового брожения по лесу Алёша набрёл на избушку бабы Яги. Он попросил помочь добраться до своего царства, на что баба Яга сказала: "Чтобы не заблудиться, старайся двигаться по маякам, которые я поставила для удобства... и вот ещё что: каждый маяк имеет свой цвет".

Баба Яга дала Алёше Поповичу коня и указала на первый маяк. И Алёша отправился в путь. В пути он обнаружил две проблемы: конь, которого дала баба Яга, оказался хромым, из-за чего мог поворачивать только на 90 градусов, и ещё коню надоедало идти в одном направлении - дойдя до маяка, он не хотел двигаться прямо.

Алёша Попович строго следовал тому, что сказала баба Яга, и всегда поворачивал на первый попавшийся по пути маяк (слева или справа от дороги). Чтобы не заблудиться во второй раз, Алёша запоминал цвета тех маяков, к которым двигался, и, после того как повернул, больше никогда не направлялся на маяк такого же цвета.

Сначала Алёша будет двигаться от избушки до основного маяка. Выясните, сможет ли Алёша добраться до своего царства.

Формат ввода:

$x_0 y_0$ координаты избушки бабы Яги ($-100 \leq x_0, y_0 \leq 100$)
 $x_i y_i$ координаты основного маяка ($-100 \leq x_i, y_i \leq 100$)
 $x y$ координаты царства ($-200 \leq x, y \leq 200$)
 n количество маяков ($0 \leq n \leq 20$)
 $x_1 y_1 c_1$ x_i, y_i - координаты маяков ($-200 \leq x_i, y_i \leq 200$)
 $x_2 y_2 c_2$ c_i - номер цвета ($1 \leq c_i \leq 255$)
 ... основной маяк имеет уникальный цвет
 $x_n y_n c_n$

Формат вывода: если Алёша доберётся до своего царства, выведите "да", иначе "нет".

Комментарий: можете полагать, что коню не придётся выбирать между левым и правым маяком.

Пример 1

input.txt

0 0
 0 3
 -3 1
 5
 -2 1 1
 -1 -4 2
 3 -2 3
 -4 1 4
 0 4 5

output.txt

да

Пример 2

input.txt

0 0
 0 3
 -3 1
 5
 -2 1 2
 -1 -4 2
 3 -2 3
 -4 1 4
 0 4 5

output.txt

нет

Пример 3

input.txt

0 0
 0 3
 -5 1
 5
 -2 1 1
 -1 -4 2
 3 -2 3

output.txt

нет

```

-4 1 4
0 4 5

```

codeforces.ru \ Тренировки
 2008-2009 Всероссийская командная олимпиада школьников по программированию (ВКОШП 08) \ В. Мост

68. Пересечение отрезка и прямой. Принадлежность точки отрезку. Пересечение отрезков. Точка внутри выпуклого многоугольника. Параметрическое уравнение прямой. Параметрическое уравнение окружности. Координаты в пространстве. Точка внутри клеточного многоугольника

Многоугольники

Автор: Дмитрий Лещенко

Петя с Васей придумали игру, в которой Вася должен был назвать число n - количество вершин выпуклого многоугольника, а Петя - назвать n вершин в порядке их обхода и ещё одну точку. Васе надо ответить на вопрос, лежит ли точка внутри многоугольника. Помогите Васе это сделать.

Точка также считается лежащей внутри многоугольника, если она лежит на одной из его сторон.

Формат ввода:

```

n                3 ≤ n ≤ 10
x1 y1
...              координаты вершин многоугольника
xn yn
x y              координаты точки.

```

Формат вывода: "да" или "нет".

input.txt

```

4
0 0
1 1
0 2
-1 1
0 1

```

output.txt

```

да

```

dl.gsu.by \ Программирование – начинающие (P/O)
 11-20 \ 2 \ 3 - "Операция поиск"

Комментарий: многоугольник может быть вырожденным в точку, то есть, возможен вариант, когда длина каждой стороны равна 0

codeforces.ru \ Тренировки

[2010-2011 Цикл интернет-олимпиад. Вторая командная олимпиада \(9 октября 2010\).](#)

[Усложненный уровень \ Е. Диаграмма](#)

Точка и отрезок

Автор: Сергей Коноплич

Определите, принадлежит ли точка отрезку.

Формат ввода: первая строка: координаты точки. Вторая и третья строки: координаты концов отрезка.

Формат вывода: "да" или "нет".

segment.in

```
0 0
1 1
3 3
```

segment.out

```
нет
```

Отрезки

Автор: Алексей Синкевич

Определите, пересекаются ли два отрезка.

Формат ввода: первые две строки содержат координаты концов первого отрезка, следующие две строки – координаты концов второго отрезка.

Формат вывода: "да" или "нет".

input.txt

```
0 0
2 1
4 2
6 3
```

output.txt

```
нет
```

Путешествие по тоннелям

Авторы: Адам Бардашевич, Сергей Кулик

Существует один город, в котором очень хорошо развита канализационная система. Жители этого города очень гордятся своей достопримечательностью. Каждый год проходит осмотр канализации. В этом году осматривать канализацию должен Влад.

Влад - очень ленивый гражданин и хочет побыстрее закончить осмотр. Влад может перемещаться по земле и под землёй – по тоннелям. Для осмотра тоннеля Влад должен пройти по земле к его началу, спуститься в тоннель, дойти до его конца и вернуться на землю.

Влад просит вас помочь ему найти кратчайший способ осмотреть канализацию, и узнать длину пути, который он должен пройти.

Канализация представлена в виде множества отрезков. Влад не очень сообразительный, и легко может заблудиться в подземных тоннелях. Он не будет осматривать пересекающиеся тоннели. Однако если тоннели лежат на одной прямой, то он без проблем сможет осмотреть такие тоннели.

Формат ввода: в первой строке находится натуральное число N - количество тоннелей. $N \leq 8$. Далее следует N строк - описание канализации. Канализация представлена в виде множества отрезков с входом в точке $p_i(x_{1i}, y_{1i})$ и выходом в точке $k_i(x_{2i}, y_{2i})$. В последней строке содержатся координаты начального положения Влада. Координаты точек не превышают 1001 по модулю.

Формат вывода: одно число - длина кратчайшего пути.

input.txt

```
4
1 1 1 2
1 3 1 6
10 10 11 11
10 11 11 10
1 1
```

output.txt

```
5.000
```

Двери

ACM Mid-central Regional Programming Contest
1996 год

Вы должны найти длину кратчайшего пути через комнату, содержащую стенки-препятствия. Комната всегда имеет стороны с $x=0$, $x=10$, $y=0$ и $y=10$. Начальная и конечная точки пути всегда (0,5) и (10,5). Также в комнате будет от 0 до 18 вертикальных стен, каждая с двумя дверными проходами.

Формат ввода: первая строка содержит количество внутренних стен. Затем для каждой такой стены будет строка, содержащая пять чисел. Первое число - координата x стены ($0 < x < 10$), оставшиеся четыре числа - y координаты краёв дверных проходов в стене.

x координаты стен будут даны в возрастающем порядке и внутри каждой строки y координаты будут также в возрастающем порядке. Входной файл будет содержать, по крайней мере, один набор данных. Конец данных будет обозначен количеством стен -1.

Формат вывода: выходной файл должен содержать одну строку вывода для каждой комнаты. Строка должна содержать минимальную длину пути, округлённую до двух десятичных знаков после запятой, и всегда должна содержать два десятичных знака после запятой. Строка не должна содержать пробелов.

doors.in

```
1
5 4 6 7 8
2
4 2 7 8 9
7 3 4.5 6 7
-1
```

doors.out

```
10.00
10.06
```

ej

2017-07-09 [Дорешивание] Тренировочная олимпиада КЮП. Контекст Вячеслава Козловского и Ивана Реева \ J. У холмов есть глаза

codeforces.ru \ Архив
1036E. Покрытые точки

Parallelepiped Walk

North-Eastern European Regional Contest

1996

Перевод: Сергей Кулик

В этой задаче вам надо найти кратчайшее расстояние между двумя точками на поверхности параллелепипеда.

Путь должен проходить по поверхности параллелепипеда. Параллелепипед имеет вершину в начале координат. Координаты любой точки, принадлежащей этому параллелепипеду неотрицательные. Ребра параллелепипеда параллельны одной из осей координат.

Формат ввода: в первой строке входного файла находятся 9 чисел: $l, w, h, x_1, y_1, z_1, x_2, y_2, z_2$. Первые три числа - длина, ширина и высота параллелепипеда. Числа x_1, y_1, z_1 - координаты первой точки, а x_2, y_2, z_2 - координаты второй. Входные данные корректны.
 $l, w, h < 1001$

Формат вывода: выведите квадрат кратчайшего расстояния между двумя точками.

input.txt

5 4 2 2 0 0 2 4 2

output.txt

36

Пиксельная графика

11 командная олимпиада школьников

Санкт-Петербург, 2008 г.

ej \ 32-08

На плоскости задан многоугольник, все стороны которого параллельны осям координат, а все вершины находятся в точках с целыми координатами.

Если провести все возможные вертикальные и горизонтальные прямые через точки с целыми координатами, то плоскость будет разбита на единичные квадраты. Необходимо найти все единичные квадраты, которые находятся внутри заданного многоугольника.

Формат ввода

Первая строка входного файла содержит три целых числа: число вершин многоугольника n ($4 \leq n \leq 100$), w и h ($1 \leq w, h \leq 200$). Каждая из последующих n строк описывает одну вершину многоугольника и содержит два целых числа x_i и y_i - ее координаты ($0 \leq x_i \leq w$, $0 \leq y_i \leq h$).

Вершины заданы в порядке обхода против часовой стрелки. Многоугольник не содержит самопересечений и самокасаний.

Формат вывода

В выходной файл выведите прямоугольник размером w на h символов, каждый из которых соответствует одному единичному квадрату. Единичный квадрат, находящийся снаружи многоугольника, обозначайте символом ".", находящийся внутри - символом "#". Левый нижний угол прямоугольника должен соответствовать точке $(0, 0)$, правый верхний - точке (w, h) .

Пример 1**pixel.in**

```
4 3 4
0 0
2 0
2 1
0 1
```

pixel.out

```
...
...
...
##.
```

Пример 2**pixel.in**

```
12 3 3
1 0
2 0
2 1
3 1
3 2
2 2
2 3
1 3
1 2
0 2
0 1
1 1
```

pixel.out

```
.#.
###
.##
```

69. Длинная арифметика

ej

2018-05-27 Тренировочная олимпиада для начинающих. Контест Александра Кривошеева \ С. $A + B = C$?

dl.gsu.by \ Олимпиады по информатике

Гомельская гор. \ 2005 \ Городская (3 декабря) \ Младшие \ 6 - "Дятлик астроном"

Х-Домино

Автор: Алексей Ропан

Вася Пупкин очень увлекается Х-домино. Однажды ему не с кем было поиграть, и он стал складывать фигурки Х-домино в виде прямоугольника размером $X \times N$, где N - количество фигурок размером $X \times 1$. Васе стало интересно - сколько существует различных вариантов сложить прямоугольник. Ответьте на этот вопрос.

Формат ввода:

$n \ x \quad 1 \leq n, x \leq 200$

Формат вывода: количество вариантов сложить прямоугольник.

Комментарий к примеру вывода: варианты расположения домино:

x\n	1	2	3	4
1	1		3	
2	2		4	

x\n	1	2	3	4
1	1		3	4
2	2			

x\n	1	2	3	4
1	1			
2	2			

x\n	1	2	3	4
1		2		
2	1		3	4

x\n	1	2	3	4
1		2		
2	1		3	4

domino.in

4 2

domino.out

5

В цифровой форме говоря

North Central Regionals

1997-98

Разработчик кроссвордов (и других подобных игр в слова) решил создать соответствие между каждым возможным словом от одного до двадцати символов и уникальными целыми числами. Соответствие очень простое - в порядке, зависящем, прежде всего, от длины слова, и затем в алфавитном порядке. Часть списка показана ниже.

a	1
b	2
...	
z	26
aa	27
ab	28
...	
snowfall	157,118,051,752
...	

Ваша работа в этой задаче - разработать программу, которая может производить двунаправленный перевод между уникальными номерами слов и соответствующими словами.

Формат ввода: ввод к программе - список слов и номеров, по одному в строке, начинающихся в столбце один, сопровождаемый строкой, содержащей единственную звездочку в столбце один. Номер будет состоять только из десятичных цифр (от 0 до 9), сопровождаемый немедленно концом строки (то есть, не будет никаких запятых во входных номерах). Слово будет состоять из от одного до двадцати строчных алфавитных символов (от a до z).

Формат вывода: вывод должен содержать единственную строку для каждого слова или номера во входных данных. Эта строка должна содержать слово, начинающееся в столбце один, сопровождаемое соответствующим количеством пробелов, и соответствующего номера слова, начинающегося в столбце 23. Номера слов, которые имеют больше чем три цифры, должны быть отделены запятыми в тысячах, миллионах, и т.д.

number.in

```
29697684282993
transcendental
28011622636823854456520
computationally
zzzzzzzzzzzzzzzzzzzz
*
```

number.out

```
elementary          29,697,684,282,993
transcendental      51,346,529,199,396,181,750
prestidigitatation  28,011,622,636,823,854,456,520
computationally     232,049,592,627,851,629,097
zzzzzzzzzzzzzzzzzzzz 20,725,274,851,017,785,518,433,805,270
```

Неравенство

Областная олимпиада по информатике

Витебск, 2001

Соавтор тестов: Владимир Миранович

Составьте программу, находящую наибольшее натуральное решение неравенства: $X! \leq B$.

$X!$ означает $1*2*3*4*...*(X-1)*X$. X и B - натуральные числа, длина B не превосходит 255 цифр.

Формат ввода: число B .

Формат вывода: искомое число.

nonequal.in

```
40322
```

nonequal.out

```
8
```

Частоты факториалов

ACM North Central Regional programming contest

1993

В попытке поддержать свой ослабевающий бизнес карманных ПК, госпожа Финикс решила предложить несколько нумерологических наработок своим клиентам. Она смогла убедить их, что частота возникновения цифр в десятичном представлении факториалов свидетельствует об их будущем. В отличие от КПК, однако, она не может вычислить в уме эти частоты, так что она наняла Вас, чтобы определить эти значения.

Считайте, что определение $n!$ (то есть, n факториал) - это $1*2*3*...*n$. Поскольку она ожидает использовать день недели, день месяца, или дня года как значение n , Вы должны быть способны определить число возникновений каждой десятичной цифры в числах такого размера, как 366 факториал ($366!$), который имеет 781 цифру.

Госпожа Финикс будет навсегда в долгу перед Вами; она могла бы даже организовать Вам путешествие, если Вы сделаете вашу работу хорошо!

Формат ввода: входные данные для программы - просто список целых чисел, для которых требуется подсчёт цифр. Все эти входные значения будут меньше или равны 366 и больше, чем 0, за исключением последнего целого числа, которое будет нулем. Не трудитесь обрабатывать это нулевое значение; просто остановите вашу программу в этом пункте.

Формат вывода: Вы должны заставить вашу программу произвести результаты, которые выглядят подобными, тем, что показаны ниже.

<i>freq.in</i>	<i>freq.out</i>									
3	3! --									
8	(0)	0	(1)	0	(2)	0	(3)	0	(4)	0
100	(5)	0	(6)	1	(7)	0	(8)	0	(9)	0
0	8! --									
	(0)	2	(1)	0	(2)	1	(3)	1	(4)	1
	(5)	0	(6)	0	(7)	0	(8)	0	(9)	0
	100! --									
	(0)	30	(1)	15	(2)	19	(3)	10	(4)	10
	(5)	14	(6)	19	(7)	7	(8)	14	(9)	20

Про шахматы

Областная олимпиада по информатике, II тур
Гомель, 10 января 2002 г.

Однажды давным-давно в далёкой стране Индии один мудрец придумал чудесную настольную игру и назвал её "шахматы". Вести о чудесной игре дошли до падишаха, и тот пожелал её увидеть. Игра ему очень понравилась (мудрец умел проигрывать), и он в порыве щедрости обещал мудрецу любую награду, какую он пожелает. Мудрец был очень хитёр и запросил то количество зерён, которое поместится на шахматной доске, если на первую клетку положить одно зёрнышко, на вторую - два, на третью - четыре и так далее (т.е. на каждую клетку нужно класть вдвое больше зерна, чем на предыдущую).

Как известно, падишах посчитал, что проще отрубить мудрецу голову, чем возиться с подсчётами...

Но позже то ли от скуки, то ли от угрызений совести падишах решил всё-таки выяснить, сколько же зерен он должен был подарить. Более того, ему стало интересно, что бы было, если бы доска имела размер N на N клеток. Так как падишах не был силен в математике, то вам придется посчитать для него размер подарка для заданного N.

Формат ввода: N - размер доски. $1 \leq N \leq 100$

Формат вывода: K - искомое количество зерен.

<i>chess.in</i>	<i>chess.out</i>
5	33554431

Удача

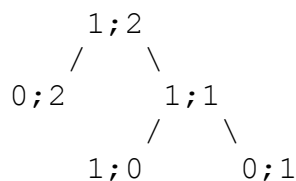
Районная олимпиада по информатике
Мозырь, 25 ноября 2000 г.

Двое друзей поспорили о том, кто из них удачливее. Чтобы проверить свою удачу, они решили сыграть в игру "Счастливчик". Правила игры таковы: у каждого из них имеется определённое количество монет. Они берут по одной монете и подбрасывают их до тех пор, пока монеты не упадут разными сторонами. После этого тот игрок, у которого выпала "решка", выводит свою

монету из игры и берёт следующую. Так продолжается до тех пор, пока у кого-то из игроков не закончатся все монеты. "Счастливым" является тот, у кого осталась хотя бы одна монета.

Ваша цель - написать программу, которая подсчитывает все возможные варианты побед, а также возможные варианты побед для первого и второго игроков.

Например, если у первого игрока одна монета, а у второго две, то все возможные случаи можно представить в следующем виде (цифры, разделённые точкой с запятой, на рисунке ниже - количество монет у первого и второго игроков соответственно):



То есть, после первого этапа игры количество своих монет у игроков может быть 0 и 2, а может быть 1 и 1; и так далее... Общее количество побед равно трём ((0;2),(1;0),(0;1)), где первый может победить в одном случае, а второй - в двух.

Формат ввода: входной файл состоит из пар чисел N и M ($1 \leq N, M \leq 100$), по одной паре в строке. Числа N и M показывают соответственно количество монет у первого игрока и у второго. Ввод заканчивается парой чисел 0 0.

Формат вывода: для каждой пары чисел N и M кроме 0 0 в выходной файл нужно вывести тройку чисел x y z, где x - количество возможных побед для первого игрока, y - для второго игрока, z - общее количество возможных побед (сумма x+y). Все числа целые.

luck.in

```

1 2
2 1
1 1
2 3
3 4
0 0
  
```

luck.out

```

1 2 3
2 1 3
1 1 2
4 6 10
15 20 35
  
```

dl.gsu.by \ Олимпиады по информатике
GCSW \ 2003 \ Начинающие \ День 1 \ 6 - "Умножение"

Длинное умножение

ACM North Central Regional Programming Contest
 1995-96

Обычно "длинным умножением" мы называем произведение двух целых x и y, полученное умножением x и отдельных цифр y, начиная с единиц. Результаты этих произведений соответствующим образом выравниваются и складываются, образуя вычисленное произведение.

Представление этих операций обычно выполняется подробным образом. Рассмотрим умножение 123 на 95:

```

123
 95
---
615
1107
-----
11685

```

Множители x и y располагаются на отдельных строках, за ними следует горизонтальная линия. Результаты умножения каждой цифры y на x затем отображаются на отдельных строках, за которыми следует ещё одна горизонтальная линия и затем конечный результат произведения. В этой задаче вы должны предоставить последовательность таких умножений, отображая результаты в традиционном представлении.

Формат ввода: каждая строка входных данных, исключая последнюю, будет содержать два целых x и y , разделённых одним или более пробелами или символами табуляции. Пустое пространство может также предшествовать первому целому и следовать за вторым целым. Каждое число будет иметь не более 100 цифр. Количество строк с числами в одном тесте не больше трёх. Последняя строка входных данных будет содержать только пустое пространство и будет признаком конца ввода.

Формат вывода: для каждой пары целых (это значит, для каждой строки за исключением последней) произведите умножение x на y , отображая результаты в форме, показанной выше, и согласно примеру, показанному ниже. После каждого умножения вставляйте в вывод пустую строку. Если y содержит только одну значащую цифру, пропустите вторую горизонтальную строку и сумму (так как в этом случае информация будет излишней). Отображайте нули только в том случае, когда они значащие.

Количество тире в первой горизонтальной линии должно соответствовать количеству цифр в большем из x и y . Количество тире во второй горизонтальной линии, если она существует, должно соответствовать количеству цифр в произведении x и y .

longmult.in

```

 4 7
135 46
 12345 862
эта строка пустая

```

longmult.out

```

 4
 7
-
28

135
 46
---
810
540
----
6210

12345
 862
-----
24690
74070
98760
-----

```


10641390

Возведение в степень

Областная олимпиада по информатике
Брест, 2001

Вычислите K^L , где $K=999N$. $0 \leq N \leq 9$.

Формат ввода:

1 строка: K

2 строка: L ($100 \leq L \leq 5000$)

Формат вывода: 1 строка: R ($R=K^L$).

Комментарий к примеру вывода: результат выводится в одну строку файла.

power.in	power.out
9991	9138941495851233417658768123038692813042881041497701849711136226403273
100	9686156676553522938160182196162144346962688606765536114161209678539445
	6095041275578667594738895899887921363719684522112967463189803788334412
	0002707932368835089966511744209422249658813736608984662260770687748114
	2680316810183499585260324770927717891006859757205067115329126506997990
	74387741977869717016860484383322161552274810044001

Число e

Областная олимпиада по информатике
Гомель, 11 января 2003 г.

Как вы знаете, знаменитое число e, которое приблизительно равно 2,7182818, может быть вычислено по следующей формуле: $E=1+1/1!+1/2!+1/3!+\dots+1/N!$. Большее N даёт более точное значение. Ваша задача - вычислить число e с точностью M точных цифр после десятичной точки.

$N! = 1*2*3*\dots*N$

Формат ввода: M - число точных цифр для вычисления ($0 < M \leq 1500$).

Формат вывода: E - вычисленное значение.

e.in	e.out
4	2.7182