

## Задача 1. Битва Магов

Условие задачи см. в отдельном файле [./Битва\\_Магов.pdf](#).

## Задача 2. Кодирование

Кодирование длин серий (Run-length encoding, RLE) – алгоритм сжатия данных, который оперирует сериями данных, то есть последовательностями, в которых один и тот же символ встречается несколько раз подряд.

При кодировании подстрока одинаковых символов, составляющих серию, заменяется строкой, которая содержит сам повторяющийся символ и количество его повторов. Таким образом, последовательность символов преобразуется в последовательность пар  $X$  и  $Y$ , где  $X$  – количество повторений символа  $Y$ . Даже если в последовательности символов указан один неповторяющийся символ, он все равно преобразуется в пару  $1Y$ .

Минимальное количество бит, которые потребуются для хранения  $X$  и  $Y$ , зависит от максимального количества символов в кодируемых строках и мощности алфавита символов, которые могут встречаться в строках соответственно.

Например, если мы знаем, что максимальная длина строки 32 символа и алфавит состоит из четырех символов, то последовательность:

AAAABBCDDD

после кодирования приобретет вид:

4A2B1C3D

и потребует для хранения 28 бит (5 бит на каждую цифру и 2 бита на каждый символ).

Пусть по указанному алгоритму кодируются строки длиной ровно 16 символов из восьми-символьного алфавита (A, B, C, D, E, F, G и H) и для хранения  $X$  и  $Y$  выбирается минимальное количество бит.

Сколько существует вариантов таких строк, отличающихся хотя бы одним символом, чтобы для хранения закодированного варианта каждой такой строки потребовалось ровно 14 бит?

## Задача 3. Отгадай-ка

Коля загадал натуральное число от 1 до  $N$ . Петя пытается отгадать это число, используя метод половинного деления, то есть задает Коле вопросы «Это число больше  $M$ ?», на которые Коля может ответить «Да» или «Нет». Число  $M$  он выбирает каждый раз так, чтобы в результате диапазон в котором на данном шаге известно, что находится искомое число, делился бы пополам. Если на очередном шаге деления в диапазоне остается нечетное количество чисел, то  $M$  выбирается таким образом, чтобы меньших чисел было на одно меньше, чем больших. Например, если  $N=9$ , то для того, чтобы отгадать число 8 потребуется 4 вопроса:

«Это число больше 4?» – «Да»;  
«Это число больше 6?» – «Да»;  
«Это число больше 7?» – «Да»;  
«Это число больше 8?» – «Нет».

Если же при том же значении  $N$  пытаться отгадать число 3, то для этого хватит всего трех вопросов:

«Это число больше 4?» – «Нет»;  
«Это число больше 2?» – «Да»;  
«Это число больше 3?» – «Нет»;

Найдите такое  $N$ , что среди всех чисел в диапазоне от 1 до  $N$  количество чисел, для отгадывания которых потребуется 11 вопросов, будет на 256 меньше количества чисел, для отгадывания которых потребуется 12 вопросов.

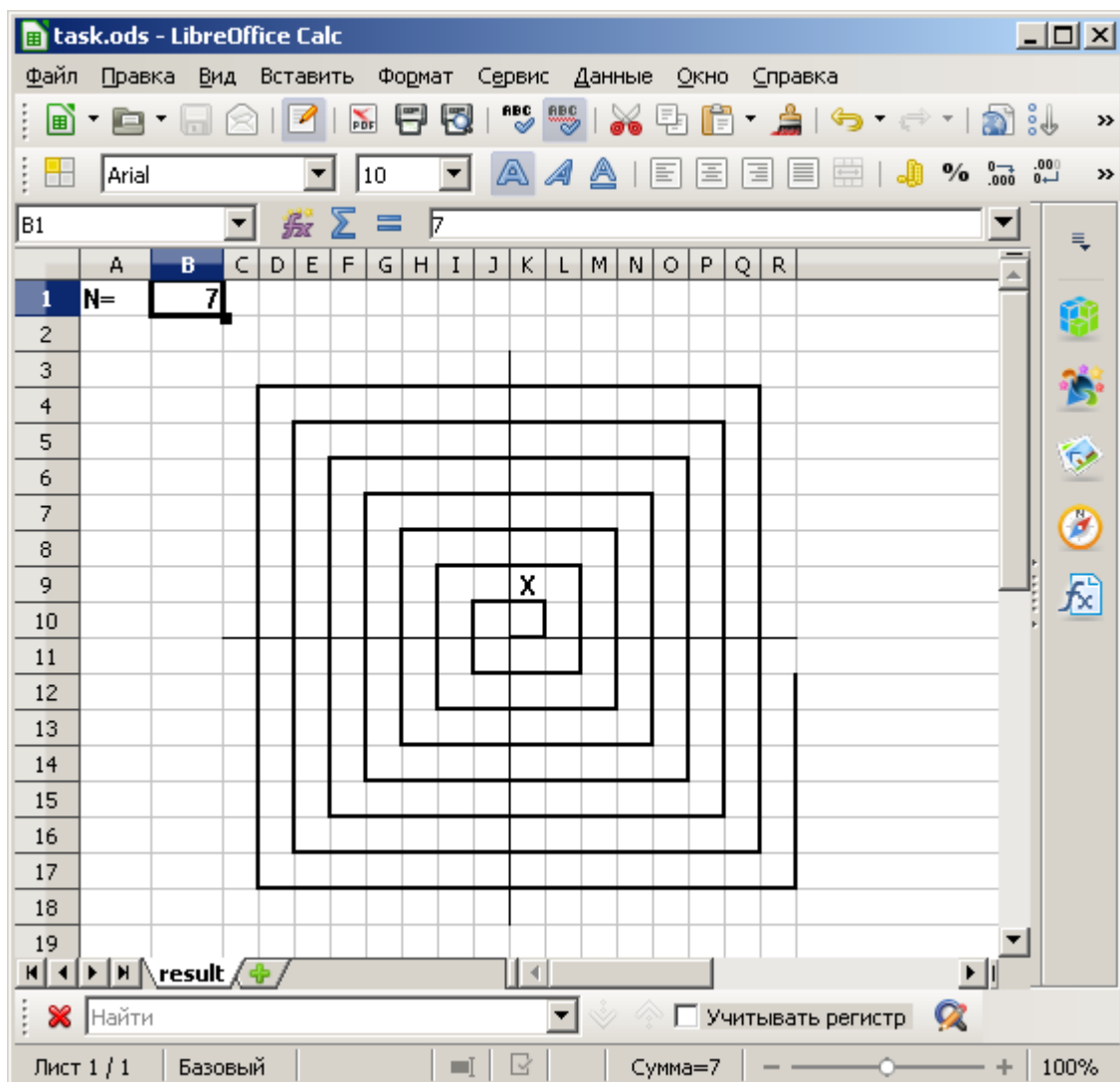
## Задача 4. Положение в лабиринте

Биологами для каких-то своих исследований с мышами был создан лабиринт в виде бесконечной спирали. Для построения лабиринта на столе возводились перегородки, соответствующие отрезкам на плоскости:  $[(0, 0), (1, 0)]$ ,  $[(1, 0), (1, 1)]$ ,  $[(1, 1), (-1, 1)]$ ,  $[(-1, 1), (-1, -1)]$ ,  $[(-1, -1), (2, -1)]$ ,  $[(2, -1), (2, 2)]$  и так далее (см. рисунок).

В экспериментах мышь помещается в начало лабиринта (клетку, левый нижний угол которой имеет координаты  $(0, 0)$ ) и бежит по лабиринту, пока биологи не сжалятся над ней и не вытащат из лабиринта. Лабиринт очень узкий, мышь не может развернуться и бежит только вперед, поворачивая в точках поворота лабиринта.

Вам необходимо смоделировать эксперимент. Для решения задачи использовать табличный процессор, например Microsoft Excel или OpenOffice/LibreOffice Calc.

Результат должен выглядеть примерно так, как показано ниже на рисунке. В ячейке B1 задается число  $N$  – общее количество перемещений мыши из клетки в клетку при движении в лабиринте (заранее известно, что мышь не может совершить более 200 перемещений). При этом буквой X автоматически отмечается положение мыши в лабиринте (см. рисунок).



Решение засчитывается, если в каждой ячейке, образующей лабиринт, записана одна и та же формула и при этом ячейки лабиринта нигде явно не пронумерованы (для нумерации

ячеек, в случае необходимости, можно использовать только одну и ту же формулу). Ячейки справа и снизу можно использовать для промежуточных вычислений.

Перегородки лабиринта можно прорисовать вручную (без использования формул).

## Задача 5. Микроконтроллер

Ученики 10А класса физико-математической школы, разбирая во время субботника стеллажи в кабинете информатики, нашли очень старый экспериментальный микроконтроллер и документацию к нему с описанием устройства микроконтроллера и его системы команд. Ради шутки (а также, чтобы заработать дополнительные баллы) несколько учеников решили выполнить довольно скучное домашнее задание по программированию на языке найденного микроконтроллера.

В такой постановке задачи из домашнего задания учеников оказались весьма интересными, поэтому вам предлагается их выполнить вместе с учениками 10А класса.

### Система команд микроконтроллера

Микроконтроллер представляет собой устройство, обладающее оперативной памятью, вычислительным ядром и интерфейсом для ввода-вывода данных (в виде целых чисел). Память для хранения инструкций программы отделена от памяти для хранения данных, поэтому далее, говоря о памяти, мы будем подразумевать память для хранения данных.

Память микроконтроллера (для хранения данных) состоит из 1000 ячеек, в каждой ячейке хранится одно целое число. Индекс первой ячейки памяти – 0, последней ячейки – 999. При запуске программы во всех ячейках памяти записано число 0.

Максимальное количество инструкций (команд) в программе – 1000, инструкции загруженной программы также нумеруются с 0. Инструкции программы микроконтроллер выполняет последовательно, начиная с первой (с индексом 0). Единственным исключением является выполнение команды условного перехода, в результате которой управление может быть передано не следующей команде, а произвольной.

Микроконтроллер прекращает свою работу в случае выполнения инструкции STP, а также при попытке выполнения любой недопустимой операции (адресация несуществующей ячейки, переход к несуществующей инструкции, деление на 0 и т.п.).

Система команд процессора содержит следующие команды:

Команда	Описание команды
IN <Arg1>	Осуществляет ввод числа с устройства ввода (консоли): <ul style="list-style-type: none"><li>&lt;Arg1&gt; – адрес (индекс) ячейки памяти, в которую будет записано введенное число</li></ul>
OUT <Arg1>	Осуществляет вывод числа с устройства ввода (консоли): <ul style="list-style-type: none"><li>&lt;Arg1&gt; – адрес ячейки, содержимое которой будет выведено на устройство вывода</li></ul>
MOV <Arg1> <Arg2>	Осуществляет запись константы в ячейку памяти: <ul style="list-style-type: none"><li>&lt;Arg1&gt; – число (константа)</li><li>&lt;Arg2&gt; – адрес ячейки, в которую будет записано значение &lt;Arg1&gt;</li></ul>
LD <Arg1> <Arg2>	Осуществляет копирование значения из ячейки, адрес которой записан в ячейке с адресом <Arg1> в ячейку с адресом <Arg2>: <ul style="list-style-type: none"><li>&lt;Arg1&gt;, &lt;Arg2&gt; – см. описание выше</li></ul> Рассмотрим пример.

	<div>Содержание памяти до выполнения команды:</div> <table><tr><td>0</td><td>0</td><td>5</td><td>0</td><td>0</td><td>10</td><td>0</td><td>0</td><td>...</td><td></td></tr></table> <div>Команда:</div> <div>LD 2 3</div> <div>Содержание памяти после выполнения команды:</div> <table><tr><td>0</td><td>0</td><td>5</td><td>10</td><td>0</td><td>10</td><td>0</td><td>0</td><td>...</td><td></td></tr></table>	0	0	5	0	0	10	0	0	...		0	0	5	10	0	10	0	0	...	
0	0	5	0	0	10	0	0	...													
0	0	5	10	0	10	0	0	...													
ST <Arg1> <Arg2>	<div>Выполняет операцию, обратную LD, т.е. осуществляет копирование значения из ячейки с адресом &lt;Arg2&gt; в ячейку, адрес которой записан в ячейке с адресом &lt;Arg1&gt;.</div> <div><ul style="list-style-type: none"><li>&lt;Arg1&gt;, &lt;Arg2&gt; – см. описание выше</li></ul></div> <div>Рассмотрим пример.</div> <div>Содержание памяти до выполнения команды:</div> <table><tr><td>0</td><td>0</td><td>5</td><td>7</td><td>0</td><td>10</td><td>0</td><td>0</td><td>...</td><td></td></tr></table> <div>Команда:</div> <div>ST 2 3</div> <div>Содержание памяти после выполнения команды:</div> <table><tr><td>0</td><td>0</td><td>5</td><td>7</td><td>0</td><td>7</td><td>0</td><td>0</td><td>...</td><td></td></tr></table>	0	0	5	7	0	10	0	0	...		0	0	5	7	0	7	0	0	...	
0	0	5	7	0	10	0	0	...													
0	0	5	7	0	7	0	0	...													
ADD <Arg1> <Arg2> <Arg3>	<div>Осуществляет операцию сложения:</div> <div><ul style="list-style-type: none"><li>&lt;Arg1&gt; – адрес ячейки, в которой записан первый аргумент для операции сложения</li><li>&lt;Arg2&gt; – адрес ячейки, в которой записан второй аргумент для операции сложения</li><li>&lt;Arg3&gt; – адрес ячейки, в которую будет помещен результат сложения</li></ul></div> <div>Примечание:</div> <div>Данную команду можно использовать для копирования значений из ячейки в ячейку. Для этого надо в качестве первого аргумента указать адрес ячейки, значение из которой необходимо скопировать, в качестве второго аргумента указать адрес ячейки, в которой записано значение 0, а в качестве третьего аргумента – адрес ячейки, в которую необходимо скопировать значение.</div>																				
SUB <Arg1> <Arg2> <Arg3>	<div>Осуществляет операцию вычитания.</div> <div>Назначение аргументов совпадает с командой ADD.</div>																				
MUL <Arg1> <Arg2> <Arg3>	<div>Осуществляет операцию умножения.</div> <div>Назначение аргументов совпадает с командой ADD.</div>																				
DIV <Arg1> <Arg2> <Arg3>	<div>Осуществляет операцию целочисленного деления.</div> <div>Назначение аргументов совпадает с командой ADD.</div>																				
JMP <Arg1> <Arg2>	<div>Осуществляет условный переход, если значение в ячейке с адресом &lt;Arg1&gt; строго больше 0.</div> <div><ul style="list-style-type: none"><li>&lt;Arg1&gt; – см. описание выше</li><li>&lt;Arg2&gt; – номер инструкции программы, которая будет выполнена следующей, если значение в ячейке с адресом &lt;Arg1&gt; больше 0</li></ul></div>																				

	Если значение в ячейке с адресом <Arg1> меньше или равно 0, то команда ничего не делает, управление передается следующей инструкции, следующей за данной.
<b>STP</b>	Останавливает работу микроконтроллера. Аргументов нет.

В программе допустимы однострочные комментарии в виде строк, начинающихся с последовательности символов "//" (см. примеры).

## Примеры программ

Ниже представлена программа ввода двух чисел, их сложений и вывода результата:

```
in 10          // ввод первого числа в ячейку памяти 10
in 11          // ввод второго числа в ячейку памяти 11

add 10 11 12   // вычисление суммы в ячейку памяти 12

out 12         // вывод результата

stp           // прекращение работы
```

Ниже представлена программа ввода последовательности (массива) чисел, и вывода той же последовательности в обратном порядке (окончанием ввода массива считается ввод отрицательного числа):

```
// [0] - 0
// [1] - 1
// [10] - a - адрес первого элемента массива
// [11] - i - индекс массива
// [12] - temp - ячейка для ввода значения
// [13] - b = a + i - ячейка для вычисления адреса элемента

mov 1 1
mov 20 10
mov 0 11      // i = 0

// :l1 = 3, первый цикл (ввод)
in 12         // temp = in()

add 12 1 12   // добавили 1 к temp (нужно для проверки в jmp)
jmp 12 7      // jmp temp :l2
jmp 1 12      // jmp :l3

// :l2 = 7
sub 12 1 12   // вернули, как было (отняли 1 от temp)
add 10 11 13  // b = a + i
st 13 12     // a[i] = temp
add 11 1 11   // i = i + 1
jmp 1 3       // jmp :l1 (безусловный переход на начало цикла)
// :l3 = 12, конец первого цикла
```

```

// здесь i указывает на элемент за последним элементом массива
// будем уменьшать i до 0
// :l4 = 12, начало второго цикла
jmp 11 14      // jmp i :l5
jmp 1 19       // jmp :l6

// :l5 = 14
sub 11 1 11     // i = i - 1
add 10 11 13    // b = a + i
ld 13 12        // temp = a[i]
out 12          // out(temp)
jmp 1 12        // jmp :l4
// :l6 = 19, конец второго цикла

stp

```

### Задание

Реализовать на языке микроконтроллера задачи из домашнего задания по программированию 10А класса.

- 1) Найти максимальное из 2-х числа. Числа вводятся с устройства ввода (консоли) и выводятся на устройство вывода (консоль).
- 2) Найти и распечатать все четные числа от 1 до N. Число N вводится с устройства ввода.
- 3) Известны A, B, C – длины сторон треугольника, необходимо определить вид треугольника. Числа A, B и C вводятся с устройства ввода (в произвольном порядке). Необходимо вывести единственное число K ( $0 \leq K \leq 3$ ): 0 – если треугольник с такими сторонами (A, B и C) не может существовать, 1 – если треугольник остроугольный, 2 – прямоугольный, 3 – тупоугольный.
- 4) Необходимо найти максимальное кол-во повторений какого-либо числа в последовательности чисел. Последовательность вводится с устройства ввода, признаком окончания ввода последовательности чисел является ввод отрицательного числа. Распечатать единственное число – максимальное кол-во повторений какого-либо числа в последовательности чисел.

### Примечание

Для запуска программы для микроконтроллера необходимо:

- Сохранить вашу программу в файле input.txt;
- Скопировать в папку с программой скрипт MICRO.bat с диска O:
- Запустить скрипт MICRO.bat из папки с программой.

## Задача 6. Популярность

Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	1

### Условие задачи

Несколько учеников 10А класса физико-математической школы поспорили, кто из них самый популярный в школе. В 10А были собраны самые способные ребята, поэтому они сразу же осознали необходимость введения терминологической базы для того, чтобы спор оставался конструктивным. После долгих обсуждений было принято следующее определение популярности:

Популярность ученика  $X$  определяется числом  $P$  – количеством учеников, которые являются друзьями ученика  $X$  или же являются друзьями друзей ученика  $X$  за исключением самого ученика  $X$ .

Ваша задача – помочь ученикам 10А установить истину в их споре.

### Входные данные

В первой строке входного файла через пробел записаны два целых числа  $N$  и  $K$  ( $1 \leq N \leq 10^3$ ,  $1 \leq K \leq 10^5$ ) – общее кол-во учеников в школе и количество пар друзей.

В каждой из следующих  $K$  строк через пробел записано по два целых числа  $X_i$  и  $Y_i$  ( $1 \leq X_i, Y_i \leq N$ ) – номера учеников, которые являются друзьями (если  $X_i$  дружит с  $Y_i$ , то  $Y_i$  также дружит с  $X_i$ ).

### Выходные данные

В первой строке выходного файла необходимо через пробел вывести два целых числа  $P$  и  $M$ , где  $P$  – максимальная популярность среди учеников школы, а  $M$  – количество учеников, имеющих максимальную популярность.

Во второй строке выходного файла через пробел необходимо вывести по возрастанию значений  $M$  целых чисел  $Z_i$  – номера учеников школы, имеющих наибольшую популярность.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
10 5 1 3 6 5 3 5 6 3 5 2	4 3 3 5 6

## Задача 7. Системы счисления

Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	1

### Условие задачи

Вам после каникул предстоит писать важную контрольную работу по информатике. Среди прочих в контрольной работе ожидаются задачи на тему «Системы счисления». Чтобы оставалось больше времени на решение других задач, вам требуется написать программу по переводу чисел между различными системами счисления.

### Входные данные

В первой строке входного файла содержится целое число  $X$  (в десятичной записи) – основание системы счисления исходного числа ( $2_{10} \leq X \leq 32_{10}$ ). Во второй строке записано само число  $N$  в указанной системе счисления ( $0 \leq N \leq 1000000_{10}$ ). В третьей строке содержится целое число  $Y$  (в десятичной записи) – основание системы счисления ( $2_{10} \leq Y \leq 32_{10}$ ), в которой вы должны представить указанное выше число  $N$ .

Цифры в системе счисления с основанием более  $10_{10}$  записываются заглавными буквами английского алфавита A, B, C, D, E, F, G, H и т.д. (в порядке следования букв латинского алфавита) для цифр со значениями 10, 11, 12, 13, 14, 15, 16, 17 и т. д.

### Выходные данные

В первой строке входного файла необходимо записать представление числа  $N$  в системе счисления  $Y$ .

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
10 5 2	101
2 11111111 17	F0
25 N 3	212
20 F0 25	C0
29 LMN 10	18322

## Задача 8. Игра «Фишки»

См. <http://c1piit03.cs.vsu.ru/06/>

Если не работает под вашим любимым браузером – открыть в Google Chrome.