

**ЗАДАНИЯ ЗАКЛЮЧИТЕЛЬНОГО  
ЭТАПА  
ИНФОРМАТИКА**

Время выполнения заданий: 240 минут

Максимальное количество баллов – 100

**Задание 1 (15 баллов). ПИРАМИДА**

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

Для строительства двумерной пирамиды используются прямоугольные блоки, каждый из которых характеризуется шириной и высотой. Можно поставить один блок на другой, только если ширина верхнего блока строго меньше ширины нижнего. Самым нижним в пирамиде может быть блок любой ширины.

По заданному набору блоков требуется определить, пирамиду какой наибольшей высоты можно построить из них.

**Формат входных данных**

В первой строке входных данных задается число  $N$  – количество блоков ( $1 \leq N \leq 100000$ ). В следующих  $N$  строках задаются пары целых чисел  $w_i$  и  $h_i$  ( $1 \leq w_i, h_i \leq 10^9$ ), разделенные пробелом – ширина и высота блока, соответственно.

**Формат выходных данных**

Целое число – максимальная высота пирамиды.

**Пример**

стандартный ввод	стандартный вывод
3 3 1 2 2 3 3	5

**Замечание**

В приведенном примере пирамида будет состоять из двух блоков: нижним будет блок с номером 3, а верхним – блок с номером 2. Блок с номером 1 нельзя использовать для строительства пирамиды, т.к. его ширина совпадает с шириной нижнего блока.

**Система оценивания**

Решения, правильно работающие при  $1 \leq N, w_i, h_i \leq 100$  будут набирать не менее 7 баллов.

**Задание 2. (15 баллов) КОЛОНИЗАЦИЯ МАРСА**

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

Ученым удалось отправить на планету Марс мини-фабрику, которая может за одни сутки произвести мини-фабрику или дрона для сбора воды (мини-фабрика или дрон для

сбора воды могут начать работу только со следующих суток). Дрон собирает одну единицу воды за одни сутки.

Определите, за какое минимальное количество суток удастся собрать не менее  $N$  единиц воды.

#### **Формат входных данных**

Задается одно число  $N$  ( $1 \leq N \leq 10^9$ ) – необходимое количество воды.

#### **Формат выходных данных**

Одно целое число  $M$  – минимально необходимое количество суток.

#### **Примеры:**

стандартный ввод	стандартный вывод
2	3

#### **Замечание**

Одна из правильных последовательностей действий выглядит так:

- В первые сутки мини-фабрика производит дрона для сбора воды;
- За вторые сутки мини-фабрика производит еще одного дрона, а первый дрон собирает одну единицу воды;
- За третьи сутки мини-фабрика может произвести еще одного дрона или мини-фабрику, при этом первый дрон собирает еще одну единицу воды (итого он собрал 2 единицы воды), а второй дрон собирает единицу воды. Таким образом, накоплено 3 единицы воды за трое суток.

Другая последовательность действий состоит в том, чтобы за первые сутки построить еще одну мини-фабрику, а за вторые сутки произвести двух дронов, которые на третьи сутки соберут 2 единицы воды.

### **Задание 3 (20 баллов). МАРШРУТЫ**

Имя входного файла: in.txt  
Имя выходного файла: out.txt  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 Мб

В государстве Чудаков  $N$  городов ( $2 \leq N \leq 16$ ), обозначаемых заглавными латинскими буквами, начиная с  $A$ , по порядку. Между некоторыми из них проложены дороги, которые могут быть как односторонними, так и двусторонними, причем не обязательно, что из каждого города можно проехать в любой другой.

В государстве всего один маршрут автобуса – 'Ч', который совершает только один рейс каждый день. Выходя из некоторого города, он совершает ровно  $N$  переездов между городами так, чтобы вернуться в тот, из которого выехал. Других ограничений на его маршрут нет. В течение дня автобус может несколько раз проехать один и тот же город или дорогу. В каждом городе существуют автобусные парки, из которых могут выезжать автобусы маршрута 'Ч'. Так что, хотя автобус каждый день возвращается в город, из которого стартовал в этот день, на следующий день начало маршрута 'Ч' может быть из любого другого города. Но рейс каждый день только один.

Маршрут обозначается  $N$  буквами, начиная с города, из которого происходит выезд. Например,  $BCDCE$  – допустимый маршрут для государства из 5 городов с соответствующими дорогами: выехать из  $B$ , проехать в  $C$ , затем в  $D$ , вернуться в  $C$ ,

проехать в Е и вернуться в изначальный город В (последний пункт маршрута, совпадающий с первым, в маршруте не указывается).

Маршрут автобуса меняется каждый день так, что список маршрутов по дням расположен в словарном порядке и содержит все возможные маршруты. Когда список кончается, его обход начинается сначала. В первый день введения маршрута 'С' автобус шёл по первому по порядку маршруту. Выведите его маршрут на день  $K$  работы маршрута.

*Пример:* В государстве четыре города: А, В, С, D. Наличие дорог между ними задано матрицей, где элемент равен 1, если из города, соответствующего строке, в город, соответствующий столбцу, есть дорога, и 0 – иначе (на главной диагонали нули – дорог, ведущих назад в тот же город, не бывает).

откуда/куда	A	B	C	D
A	0	0	1	1
B	1	0	1	1
C	0	1	0	0
D	0	1	1	0

Полное расписание маршрутов в таком государстве выглядит так:

ADCB  
 BADC  
 BCBC  
 BCBD  
 BDBC  
 BDBD  
 CBAD  
 CBCB  
 CBDB  
 DBCB  
 DBDB  
 DCBA

Таким образом, например, маршрут на день 30 – это BDBD.

### **Формат входных данных**

В первой строке указывается количество городов  $N$  ( $2 \leq N \leq 16$ ). Далее следует  $N$  строк по  $N$  элементов (цифр), разделенных пробелом, содержащих матрицу, задающую дороги между городами. Далее следует строка содержащая целое число  $D$  – номер дня, маршрут которого требуется определить ( $1 \leq D \leq 2^{64}$ ).

### **Формат выходных данных**

В единственной строке указывается маршрут, т.е. порядок посещения городов, например BDBD (см. предыдущий пример).

### **Пример**

in.txt	out.txt
3 0 1 1 1 0 1 1 1 0 4	BCA

### **Система оценивания**

Пять тестов. В зависимости от сложности за каждый правильный ответ от 2 до 6 баллов.

#### Задание 4 (20 баллов) ШИФРОВАНИЕ

Имя входного файла: in.txt  
Имя выходного файла: out.txt  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 Мб

Одним из самых простых способов шифрования открытого текста является шифр простой замены. Он состоит в том, что каждая буква в алфавите, которым написано открытое сообщение, заменяется на какой-то другой символ, например, другую букву того же алфавита. Пусть дана таблица замены, использующая для замены только 33 буквы русского алфавита в верхнем регистре (заглавные буквы):

Сообщение	Шифртекст	Сообщение	Шифртекст	Сообщение	Шифртекст
А	Г	К	Т	Х	З
Б	Ш	Л	Х	Ц	Ж
В	Ы	М	Я	Ч	Л
Г	О	Н	Ь	Ш	Ё
Д	Э	О	Ф	Щ	Н
Е	Ц	П	У	Ъ	Д
Ё	М	Р	К	Ы	Е
Ж	Ъ	С	Ю	Ь	Б
З	Щ	Т	Р	Э	Ч
И	А	У	П	Ю	И
Й	В	Ф	С	Я	Й

Если применить замену, заданную такой таблицей, к слову «ДОМ», получится зашифрованный текст «ЭФЯ». Если применить замену к полученному результату, из «ЭФЯ» получится «ЧСЙ», а из «ЧСЙ» таким способом можно получить текст «ЛЮВ».

Известно, что через некоторое количество применений замены полученный результат совпадет с исходным словом «ДОМ», после чего результаты замены начнут повторяться.

Определите, сколько различных шифртекстов (включая совпадающий с исходным словом) можно получить из произвольного заданного слова по произвольно заданной таблице замены таким способом.

#### **Рекомендации.**

До начала работы над программной реализацией постарайтесь найти ответы на следующие вопросы:

1. Сколько различных зашифрованных текстов (включая и совпадающий с открытым текстом) можно получить одной операцией замены из открытого текста с  $n$  различными буквами, используя все возможные таблицы замены.
2. Можно ли получить все возможные зашифрованные тексты (число которых установлено в пункте 1), применяя к результату зашифрования операцию замены символов по одной и той же таблице неограниченное число раз.

#### **Формат входного файла:**

В первой строке файла задана последовательность заглавных букв, заменяющих буквы, стоящие в алфавитном порядке (таблица замены). Например, приведенной выше таблице соответствует строка «ГШЫОЭЦМЪЩАВТХЯЬФУКЮРПСЗЖЛЁНДЕБЧИЙ». В следующей строке задано слово, являющееся открытым текстом – в верхнем регистре (заглавными буквами) без пробелов. Например, слово «КРИПТОАНАЛИЗ».

Каждая из этих строк заканчивается либо символами с кодами 13, 10 (окончание строк DOS – для Pascal ABC .NET), либо символом с кодом 10 (окончание строк Unix) в зависимости от выбранного при сдаче программы типа конца строк. Никаких других символов в двух входных строках не встречается.

Русский текст задан в кодировке Windows-1251 (cp1251). В ней заглавные русские буквы от "А" до "Я" кроме буквы "Ё" имеют коды от 192 (шестнадцатеричное C0) до 223 (шестнадцатеричное DF). Буква "Ё" имеет код 168 (шестнадцатеричное A8). Русские буквы (кроме "Ё") упорядочены по алфавиту.

**Формат выходного файла:**

В единственной строке выведите число, соответствующее количеству различных возможных шифртекстов, которые можно получить из заданного открытого текста с помощью заданной таблицы замены.

**Примеры:**

in.txt	out.txt
ГШЫОЭЦМЪЩАВТХЯЬФУКЮРПСЗЖЛЁНДЕБЧИЙ КРИПТОАНАЛИЗ	42
УКЮРПСЗЖЛЁНДЕБЧИЙГШЫОЭЦМЪЩАВТХЯЬФ СВЕРХСЕКРЕТНО	154

**Система оценивания:**

10 тестов. Корректно работающая программа для каждого из тестов: +2 балла.

**Задание 5 (30 баллов) СКОБКИ**

Имя входного файла: in.txt  
 Имя выходного файла: out.txt  
 Ограничение по времени: 1 сек  
 Ограничение по памяти: 256 Мб

Дано корректное математическое выражение, состоящее из переменных, обозначаемых строчными латинскими буквами, инфиксных бинарных операций и круглых скобок для группировки подвыражений.

Все операции имеют ассоциативность слева направо и приоритеты, указанные в таблице:

Приоритет	Операции
1 (наибольший)	*, /
2	+, —
3	&
4	^
5 (наименьший)	

Требуется удалить из выражения все лишние пары скобок, не влияющие на порядок операций в нём (операции трактовать абстрактно, без какого-либо математического смысла, опираясь только на формальный порядок операций).

Приоритет определяет, в каком порядке выполняются операции в цепочке, а ассоциативность определяет направление вычислений в цепочке операций одного приоритета.

**Примеры:**

ВХОД	ВЫХОД
$a+(b*c)$	$a+b*c$

(у ‘\*’ приоритет выше, чем у ‘+’, поэтому она и так выполняется первой,- скобки лишние);

ВХОД	ВЫХОД
$((a+b)+(c+d))$	$a+b+(c+d)$

(Скобки вокруг всего выражения допустимы, но никогда не влияют на порядок вычисления внутри. Поскольку ассоциативность всех операций слева направо, первые внутренние скобки лишние, а вторые – нет, без них выражение было бы эквивалентно  $((a+b)+c)+d$ );

ВХОД	ВЫХОД
$((a+b)&c^d)$	$(a+b)&c^d$

(скобки вокруг переменной всегда лишние).

in.txt	out.txt
$((a)&b^c ((d)))$	$a&b^c d$
a	a

**Формат входного файла:**

Одна строка, содержащая исходное математическое выражение не длиннее 100 символов.

**Формат выходного файла:**

Одна строка с математическим выражением без лишних скобок.

**Система оценивания**

Всего 10 тестов. Корректно работающая программа для каждого из тестов: +3 балла.