



ВАРИАНТ 1

Задача 1 (5 баллов)

Разведчики

От разведчиков были получены сведения, закодированные в виде электронной таблицы. Определите числа в ячейках C2:C11, которые будут получены в результате копирования в них блока ячеек B1:B10.

В ячейках A1:A5 разместите десятичные результаты вычисления чисел разных систем счисления:

$$A1 = 21_5 - 12_3 + 10_3$$

$$A2 = 12_4 - 11_3 + 10_2$$

$$A3 = 20_5 - 12_3 + 10_4$$

$$A4 = 11_3 - 11_2 + 10_5$$

$$A5 = 20_4 - 12_5 + 11_5$$

В ячейке A6 разместите формулу: ="АБВ".

	A	B	C	D
1		=СУММ(A\$1:\$A6)		
2		=МИН(\$A\$1:A\$6;B\$1)		
3		=МАКС(\$A1:A\$6;B\$1:B2)		
4		=ЕСЛИ(A\$6=0;A\$4;A\$5)		
5		=СУММЕСЛИ(A1:B\$4;A2)		
6		=СЧЁТЕСЛИ(A1:B\$5;"<A4")		
7		=ЕСЛИ(ИЛИ(B\$3>\$B4;\$B\$5<\$B6);A4;A\$5)		
8		=ЕСЛИ(И(B4<\$B5;\$B6>\$B7);A1;A2)		
9		=ЕСЛИ(B7-\$B\$8>0;B\$7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;B5))		
10		=ЕСЛИ(B8>\$B\$9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))		
11				
12				

Ответ записать в одну строчку через запятую, начиная с ячейки C2 и заканчивая ячейкой C11.

Решение

Получим значения в ячейках A1:A5 как результат вычисления чисел разных систем счисления:

$$A1 = 21_5 - 12_3 + 10_3 = 11 - 5 + 3 = 9$$

$$A2 = 12_4 - 11_3 + 10_2 = 6 - 4 + 2 = 4$$

$$A3 = 20_5 - 12_3 + 10_4 = 10 - 5 + 4 = 9$$

$$A4 = 11_3 - 11_2 + 10_5 = 4 - 3 + 5 = 6$$

$$A5 = 20_4 - 12_5 + 11_5 = 8 - 7 + 6 = 7$$

Следующим шагом копируем блок ячеек B1:B10 в ячейки C2:C11, в результате получим следующие формулы:



	A	B	C
1	9	=СУММ(A\$1;\$A6)	
2	4	=МИН(\$A\$1:A\$6;B\$1)	=СУММ(B\$1;\$A7)
3	9	=МАКС(\$A1:A\$6;\$B\$1:B2)	=МИН(\$A\$1:B\$6;C\$1)
4	6	=ЕСЛИ(A\$6=0;A\$4;\$A5)	=МАКС(\$A2:B\$6;\$B\$1:C3)
5	7	=СУММЕСЛИ(A1:B\$4;A2)	=ЕСЛИ(B\$6=0;B\$4;\$A6)
6	= "АБВ"	=СЧЁТЕСЛИ(A1:B\$5;"<A4")	=СУММЕСЛИ(B2:C\$4;B3)
7		=ЕСЛИ(ИЛИ(B\$3>\$B4;\$B\$5<\$B6);A4;\$A5)	=СЧЁТЕСЛИ(B2:C\$5;"<A4")
8		=ЕСЛИ(И(B4<\$B5;\$B6>\$B7);A1;A2)	=ЕСЛИ(ИЛИ(C\$3>\$B5;\$B\$5<\$C6);B5;\$B5)
9		=ЕСЛИ(B7-\$B\$8>0;\$B7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;\$B5))	=ЕСЛИ(И(C5<\$C6;\$C7>\$C8);B2;\$B3)
10		=ЕСЛИ(B8>\$B9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))	=ЕСЛИ(C8-\$B\$8>0;\$C7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;\$B5))
11			=ЕСЛИ(C9>\$C9;СЧЁТЕСЛИ(C2:C10;">7");СЧЁТЕСЛИ(C2:C10;"<7"))
12			

Вычисляем значения по формулам в ячейках B1:B10, затем переходим к вычислениям значений в ячейках C2:C11. Получаем следующие результаты:

	A	B	C	D
1	9	9		
2	4	4	9	
3	9	9	0	
4	6	7	9	
5	7	8	7	
6	АБВ	0	27	
7		6	0	
8		4	8	
9		6	9	
10		5	0	
11			3	
12				

Ответ: 9, 0, 9, 7, 27, 0, 8, 9, 0, 3

**Задача 2 (10 баллов)****Ключ от архива**

Историк Петр обнаружил старый архив, защищённый кодовым замком. Документы содержат информацию, зашифрованную четырехразрядным числовым ключом в шестнадцатеричной системе счисления. Петр знает, что каждая цифра числа слева направо была преобразована в двоичную систему счисления и записана четырехразрядным равномерным кодом, т.е. если двоичное представление содержало менее четырёх разрядов, оно дополнялось нулями слева до полного четырёхразрядного кода (например, $716=01112$). Полученные двоичные коды соединили последовательно слева направо.

Таблица истинности этого логического выражения содержит все необходимые значения в результирующем столбце сверху вниз для определения кода:

$$a \wedge b \equiv (c \vee \neg d) \rightarrow (c \oplus b \wedge a \wedge d) \equiv (b \vee a \wedge d)$$

Строки значений a, b, c, d в таблице истинности идут последовательно, в лексикографическом порядке 0000, 0001, 0010 и т.д. Определите четырёхзначный код архива в шестнадцатеричной системе счисления.

Решение

Составим таблицу истинности

a	b	c	d	$a \wedge b$	$\neg d$	$c \vee \neg d$	$a \wedge b \equiv c \vee \neg d$	$b \wedge a$	$b \wedge a \wedge d$	$c \oplus b \wedge a \wedge d$	$a \wedge b \equiv (c \vee \neg d) \rightarrow (c \oplus b \wedge a \wedge d)$	$a \wedge d$	$b \vee a \wedge d$	Итог
0	0	0	0	0	1	1	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1
0	0	1	0	0	1	1	0	0	0	1	1	0	0	0
0	0	1	1	0	0	1	0	0	0	1	1	0	0	0
0	1	0	0	0	1	1	0	0	0	0	1	0	1	1
0	1	0	1	0	0	0	1	0	0	0	0	0	1	0
0	1	1	0	0	1	1	0	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0	0	1	1	0	1	1
1	0	0	0	0	1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0	1	1	0
1	0	1	0	0	1	1	0	0	0	1	1	0	0	0
1	0	1	1	0	0	1	0	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	0	0	0	0	1	0
1	1	0	1	1	0	0	0	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	0	1	1	0	1	1
1	1	1	1	1	0	1	1	1	1	0	0	1	1	0

Двоичный код в последнем столбце можно разбить на четыре кодовых слова длины 4 числа следующим образом: 0100, 1011, 0001, 0110.

Таким образом получаем следующий шестнадцатеричный код: $4(11)16_{16}$ или $4B16_{16}$

Ответ: $4(11)16_{16}$ или $4B16_{16}$



Задача 3 (15 баллов)

Циклический возврат камеры наблюдения

Исполнитель Камера используется для управления камерой наблюдения, которая может вращаться ограниченно по вертикали и без ограничений по горизонтали. Исполнитель Камера может выполнять следующие команды:

1) Повернуть на (a, b) , где a, b – целые числа. Эта команда поворачивает камеру на величину вверх по вертикали на угол a , заданный в градусах, и поворачивает его против часовой стрелки на угол b , заданный в градусах. Например, команда Повернуть на $(-2, -30)$ опустит камеру на 2 градуса по вертикали вниз и повернет его на угол 30 по горизонтали по часовой стрелке.

2) Цикл:

ПОВТОРИ n

последовательность команд

КОНЕЦ ПОВТОРИ

Этот цикл означает, что последовательность команд будет выполняться n раз.

Исполнителю Камера был дан для исполнения следующий алгоритм (буквами a, b и n обозначены неизвестные целые числа):

НАЧАЛО

повернуть на $(50, 30)$

ПОВТОРИ n

повернуть на (a, b)

повернуть на $(6, -30)$

КОНЕЦ ПОВТОРИ

повернуть на $(0, 180)$

КОНЕЦ

Чему равно максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов. Чему равно минимально возможное число полных оборотов (360 градусов), которое при этом совершит камера?

Чему равны значения n , если после выполнения алгоритма камера вернулась в то же состояние, что было до начала выполнения алгоритма, совершив 5 полных оборотов? Чему, при этом равны параметры a и b ?

Решение

Значение параметров исполнителя Камера в результате выполнения алгоритма рассчитывается по формуле:

$$l - l_0 = 50 + n(a + 6) + 0$$



$$f - f_0 = 30 + n(b - 30) + 180$$

В этих уравнениях n – количество повторений команд ПОВТОРИ, l и l_0 – конечное и начальное значение угла вертикального поворота камеры, f и f_0 – конечное и начальное значение угла горизонтального поворота камеры.

Так как камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, $l - l_0 = 0$, а $f - f_0 = 360k$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Знак $+$ соответствует оборотам против часовой стрелки, а знак $-$ соответствует оборотам по часовой стрелке.

С учетом этих условий, получим систему уравнений:

$$50 + n(a + 6) + 0 = 0$$

$$30 + n(b - 30) + 180 = 360k$$

После преобразований, получим:

$$n(a + 6) = -50$$

$$n(b - 30) = 360k - 210$$

Нужно найти натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число n должно быть одновременно делителем чисел 50 и $360k - 210$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Наибольшим общим делителем для этих выражений будет число 50. Оно будет соответствовать максимально возможному значению n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, при условии, что $360k - 210$ будет кратно 50. Минимальное число оборотов равно 1, что соответствует $k = 1$.

При совершении 5 полных оборотов возможны два варианта, в зависимости от направления вращения:

$$n(b - 30) = 360 \cdot 5 - 210 = 1590$$

$$n(b - 30) = 360 \cdot (-5) - 210 = -2010$$

Нужно найти наибольшее натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b – наибольшее число, которое является делителем чисел 50 и 1590 или 50 и 2010. В обоих случаях наибольший общий делитель $n = 10$. Рассчитав значения a и b , получим:

$$a = -50/10 - 6 = -11$$

$$b- = -2010/10 + 30 = -171 \text{ – при оборотах по часовой стрелке}$$

$$b+ = 1590/10 + 30 = 189 \text{ – при оборотах против часовой стрелки}$$

Остальные значения n представляют собой делители числа 10: 1, 2, 5. Для них расчет значений a и b производится по аналогичным формулам.

Ответ: максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 50, при этом минимально возможное число полных оборотов равно 1. При совершении 5 полных оборотов максимальное $n = 10$, $a = -11$, $b = -171$



при оборотах по часовой стрелке и $b = 189$ при оборотах против часовой стрелки.

**Задача 4 (20 баллов)****Геологический шифр**

В ходе геологоразведочных работ в Арктической зоне РФ была обнаружена неизвестная науке горная порода с уникальными кристаллическими свойствами. Чтобы сохранить приоритет открытия в тайне от конкурентов, старший геолог Пётр передал для анализа коллеге Марии зашифрованные данные, содержащие предварительное название породы из двух слов.

Мария получила два файла:

- 1) rock_sample.dat – файл с основным зашифрованным отчётом.
- 2) thin_section.png – микрофотография тонкого среза породы, в которую Петром был внедрён секретный ключ методом стеганографии, имитирующим природные включения.

Алгоритм действий для Марии выглядит следующим образом:

1) Извлечь скрытые данные из изображения thin_section.png. Для этого необходимо проанализировать младшие биты цветовых каналов пикселей, что позволит получить битовую последовательность.

2) Эту битовую последовательность необходимо декодировать. В результате она должна превратиться в строку в формате JSON без кавычек, которая и содержит секретные ключи для алгоритма RSA. В данной задаче строка в формате JSON без кавычек выглядит следующим образом: {ключ1:значение1,ключ2: значение2,ключ3:значение3}.

3) Используя полученные ключи, можно приступить к дешифровке основного отчёта, хранящегося в файле rock_sample.dat.

В результате анализа младших битов цветовых каналов пикселей изображения thin_section.png Мария получила следующую строку:

11000000111001111110110100100001011001111101010011101000001010
110011111001111011110001

Пётр, известный своей любовью к горным треккигам, для защиты информации в файле thin_section.png применяет специальный «Горный код» стеганографии. Его принцип основан на использовании неравномерного кодирования, где частым символам соответствуют короткие коды, что минимизирует общий размер секретного сообщения. Дополнительный уровень защиты обеспечивают «мусорные» символы – специальные ловушки, встроенные в код. При расшифровке легитимный получатель должен распознать и отфильтровать этот мусор, чтобы получить исходные данные. Таблица используемых Петром кодов представлена ниже.

Символ	Код	Символ	Код
{	110000	7	111100
}	110001	8	111101
"	110010	9	111110
:	110011	n	000



Символ	Код	Символ	Код
,	110100	e	001
0	110101	d	010
1	110110	p	0110
2	110111	q	0111
3	111000	x	1000
4	111001	z	1001
5	111010	y	1010
6	111011		

Неравномерный код – это код, в котором кодовые слова, например, последовательности битов, имеют разную длину.

Например, при декодировании битовой последовательности 000001010 с помощью вышепредставленной таблицы, процесс декодирования будет заключаться в следующем:

1) Читаем биты слева-направо: 0 → 00 → 000.

000 есть в таблице? Да, это n.

Результат: n.

2) Продолжаем со следующего бита: 0 → 00 → 001.

001 есть в таблице? Да, это e.

Результат: e.

3) Продолжаем со следующего бита: 0 → 01 → 010.

010 есть в таблице? Да, это d.

Результат: d.

Расшифрованное сообщение n e d.

Файл rock_sample.dat представляет собой последовательность десятичных чисел, каждая из которых – зашифрованная буква передаваемого сообщения. Исходный текст отчёта перед шифрованием был преобразован в числа по правилу: каждый символ кодировался своим порядковым номером в алфавите (a=1, б=2, ..., я=33). Для шифрования Петром использовалась формула: $C_i = M_i^e \bmod n$,

где C_i – i-й элемент последовательности шифротекста C;

M_i – i-й элемент последовательности шифра M;

e – открытая экспонента – параметр открытого ключа RSA – число, используемое при шифровании в качестве степени;

n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.

Для дешифрования должна использоваться формула $M_i = C_i^d \bmod n$,

где M_i – i-й элемент последовательности шифра M;

C_i – i-й элемент последовательности шифротекста C;

d – секретная экспонента – параметр закрытого ключа RSA – число, используемое при дешифровке в качестве степени;

n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.



Совокупность значений (e, n) и (d, n) является открытым и закрытым ключами, используемыми при шифровании и дешифровании соответственно.

Например, RSA-шифрование с открытым ключом: $(e, n) = (17, 3233)$ – и закрытым ключом: $(d, n) = (2753, 3233)$ заключается в следующем.

При шифровании сообщения $M = 65$ отправляемый шифротекст $C = M^e \bmod n = 65^{17} \bmod 3233 = 2790$

При дешифровке полученного шифротекста $C = 2790$ восстановленное сообщение $M = C^d \bmod n = 2790^{2753} \bmod 3233 = 65$.

Примечание

Чтобы вычислить выражение вида $M = C^d \bmod n$, иногда невыгодно сначала находить большое число C^d , ведь оно может быть огромным. Удобнее использовать алгоритм быстрого возведения в степень по модулю.

Идея алгоритма:

1. Разложить показатель степени на степени двойки.
2. Последовательно возводить число в квадрат.
3. После каждого шага брать остаток по модулю.
4. Умножать только те степени, которые нужны по разложению.

Например, если нужно найти значение выражения $11^5 \bmod 9$, то степень 5 можно представить как сумму степеней 4 и 1: $5 = 4 + 1$.

Следовательно:

1. Находим остаток: $11^2 \bmod 9 = 4$, т.к. $9 \times 13 = 117$ (остаток 4).

2. Используем найденный остаток, чтобы получить $11^4 \bmod 9$.

$11^4 \bmod 9 = (11^2)^2 \bmod 9 \equiv 4^2 \bmod 9 = 7$, т.к. $9 \times 1 = 9$ (остаток 7)

3. Перемножаем найденный остаток с 11 – и снова берём остаток.

$11^5 \bmod 9 = (11^4 \times 11) \bmod 9 \equiv (7 \times 11) \bmod 9 = 77 \bmod 9 = 5$,
т.к. $9 \times 8 = 72$ (остаток 5)

Таким образом, $11^5 \bmod 9 = 5$.

Используя данные, извлечённые из стеганографического сообщения, расшифруйте rock_sample.dat и узнайте предварительное название новой горной породы. Содержимое rock_sample.dat:

Первое слово – 33 20 25 39 19; второе слово – 4 8 37 36 3 8 44 16 12.

В ответе укажите два слова расшифрованного сообщения в формате «Первое слово» «Второе слово» (например, Медное золото).

Решение

На первом этапе необходимо определить секретный ключ, который был внедрен Петром методом стеганографии в файл thin_section.png.

В задаче сказано, что Пётр использовал для шифрования специальный вод, который основан на использовании неравномерного кодирования. Алгоритм дешифровки в данном случае заключается в следующем: читаем строку по битам слева направо, пошагово накапливая последовательность,



пока она не совпадёт с одним из кодов в таблице. Как только совпадение найдено, записываем символ и начинаем формировать следующую последовательность заново.

В условии сказано, что Мария в результате анализа данного изображения получила битовую строку:

1100000011100111111101101001000010110011111010100111010000010
10110011111001111011110001

Начнём процесс декодирования. Если код не найден сразу (например, при чтении первой 1), то показан процесс накопления последовательности: $1 \rightarrow 11 \rightarrow 110$ и т.д., до момента совпадения с записью в таблице кодов, используемых Петром при шифровании. Максимальная длина кодового слова в таблице кодировки – 6 бит (например, 110000, 110011, 111110 и т.д.), поэтому в процессе накопления битовой последовательности последовательности длиннее 6 символов не рассматриваются.

Шаг	Индекс начала	Биты (извлечённые)	Найденный код	Символ
1	0	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110000$	110000	{
2	6	$0 \rightarrow 00 \rightarrow 001$	001	e
3	9	$110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
4	15	$111 \rightarrow 1111 \rightarrow 11111 \rightarrow 111110$	111110	9
5	21	$1101 \rightarrow 11010 \rightarrow 110100$	110100	,
6	27	$1 \rightarrow 10 \rightarrow 100 \rightarrow 1000$	1000	x
7	31	$0 \rightarrow 01 \rightarrow 010$	010	d
8	34	$11 \rightarrow 110 \rightarrow 1100 \rightarrow 110011$	110011	:
9	40	$111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111010$	111010	5
10	46	$1 \rightarrow 10 \rightarrow 100 \rightarrow 1001$	1001	z
11	50	$1101 \rightarrow 11010 \rightarrow 110100$	110100	,
12	56	$0 \rightarrow 00 \rightarrow 000$	000	n
13	59	$1 \rightarrow 10 \rightarrow 101 \rightarrow 1010$	1010	y
14	63	$11 \rightarrow 110 \rightarrow 1100 \rightarrow 110011$	110011	:
15	69	$111 \rightarrow 1110 \rightarrow 11100 \rightarrow 111001$	111001	4
16	75	$111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111011$	111011	6
17	81	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110001$	110001	}

Собирая полученные символы в порядке их дешифровки, получаем точную расшифровку битовой строки: {e:9,xd:5z,ny:46}. В условии задачи говорится про «мусорные символы» – ловушки, которые Пётр вставляет в сообщение. В таблице кодов есть набор букв-«включений» x, y, z (и т.д.), и



они выступают как мусор в конкретном сообщении: они появляются внутри ключей/значений, но не несут смысла для конечного JSON.

Если удалить все мусорные буквы x, y, z из полученной строки {e:9,xd:5z,ny:46}, то получим корректную JSON-подобную структуру без кавычек: {e:9,d:5,n:46}. Следовательно, получены данные для расшифровки текстового сообщения из файла rock_sample.dat, а именно модуль n=46 и секретная экспонента d=5.

Файл rock_sample.dat содержит две последовательности целых чисел, где каждое число – зашифрованный символ шифротекста:

Первое слово: 33 20 25 39 19

Второе слово: 4 8 37 36 3 8 44 16 12

Из условия задачи воспользуемся формулой дешифрования: $M_i = C_i^d \bmod n$,

где M_i – i-й элемент последовательности шифра M (восстановленный номер буквы (1 = а, 2 = б, ..., 33 = я);

C_i – i-й элемент последовательности шифротекста C;

d – секретная экспонента;

n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.

Дешифруем первое слово: C = [33 20 25 39 19]. Можно производить дешифровку напрямую возводя числа в степень d=5 или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 33$

$$33^2 = 1089 \rightarrow 1089 \bmod 46 = 31, \text{ т. к. } 46 \times 23 = 1058 \text{ (остаток 31)}$$

Таким образом, $33^2 \equiv 31 \pmod{46}$

$$33^4 = (33^2)^2 \equiv 31^2 = 961 \rightarrow 961 \bmod 46 = 41, \text{ т. к. } 46 \times 20 = 920 \text{ (остаток 41)}$$

Таким образом, $33^4 \equiv 41 \pmod{46}$

$$33^5 = 33^4 \times 33 \equiv 41 \times 33 = 1353 \rightarrow 1353 \bmod 46 = 19, \text{ т. к. } 46 \times 29 = 1334 \text{ (остаток 19)}$$

Таким образом, $M_1 = 33^5 \pmod{46} = 19$

Следовательно, $M_1 = 19$, что соответствует букве алфавита «с».

2. $C_2 = 20$

$$20^2 = 400 \rightarrow 400 \bmod 46 = 32, \text{ т. к. } 46 \times 8 = 368 \text{ (остаток 32)}$$

Таким образом, $20^2 \equiv 32 \pmod{46}$

$$20^4 = (20^2)^2 \equiv 32^2 = 1024 \rightarrow 1024 \bmod 46 = 12, \text{ т. к. } 46 \times 22 = 1012 \text{ (остаток 12)}$$

Таким образом, $20^4 \equiv 12 \pmod{46}$

$$20^5 = 20^4 \times 20 \equiv 12 \times 20 = 240 \rightarrow 240 \bmod 46 = 10, \text{ т. к. } 46 \times 5 = 230 \text{ (остаток 10)}$$

Таким образом, $M_2 = 20^5 \pmod{46} = 10$

Следовательно, $M_2 = 10$, что соответствует букве алфавита «и».

3. $C_3 = 25$



$$25^2 = 625 \rightarrow 625 \bmod 46 = 27, \text{ т. к. } 46 \times 13 = 598 \text{ (остаток 27)}$$

Таким образом, $25^2 \equiv 27 \pmod{46}$

$$25^4 = (25^2)^2 \equiv 27^2 = 729 \rightarrow 729 \bmod 46 = 39, \text{ т. к. } 46 \times 15 = 690 \text{ (остаток 39)}$$

Таким образом, $25^4 \equiv 39 \pmod{46}$

$$25^5 = 20^4 \times 25 \equiv 39 \times 25 = 975 \rightarrow 975 \bmod 46 = 9, \text{ т. к. } 46 \times 21 = 966 \text{ (остаток 9)}$$

Таким образом, $M_3 = 25^5 \pmod{46} = 9$

Следовательно, $M_3 = 9$, что соответствует букве алфавита «з».

4. $C_4 = 39$

$$39^2 = 1521 \rightarrow 1521 \bmod 46 = 3, \text{ т. к. } 46 \times 33 = 1518 \text{ (остаток 3)}$$

Таким образом, $39^2 \equiv 3 \pmod{46}$

$$39^4 = (39^2)^2 \equiv 3^2 = 9$$

$$39^5 = 39^4 \times 39 \equiv 9 \times 39 = 351 \rightarrow 351 \bmod 46 = 29, \text{ т. к. } 46 \times 7 = 322 \text{ (остаток 29)}$$

Таким образом, $M_4 = 39^5 \pmod{46} = 29$

Следовательно, $M_4 = 29$, что соответствует букве алфавита «ы».

5. $C_5 = 19$

$$19^2 = 361 \rightarrow 361 \bmod 46 = 39, \text{ т. к. } 46 \times 7 = 322 \text{ (остаток 39)}$$

Таким образом, $19^2 \equiv 39 \pmod{46}$

$$19^4 = (19^2)^2 \equiv 39^2 = 1521 \bmod 46 = 3, \text{ т. к. } 46 \times 33 = 1518 \text{ (остаток 3)}$$

Таким образом, $19^4 \equiv 3 \pmod{46}$

$$19^5 = 19^4 \times 19 \equiv 3 \times 19 = 57 \rightarrow 57 \bmod 46 = 11, \text{ т. к. } 46 \times 1 = 46 \text{ (остаток 11)}$$

Таким образом, $M_5 = 19^5 \pmod{46} = 11$

Следовательно, $M_5 = 11$, что соответствует букве алфавита «й».

Соединив расшифрованные буквы, получаем слово «сизый».

Дешифруем второе слово: $C = [4 \ 8 \ 37 \ 36 \ 3 \ 8 \ 44 \ 16 \ 12]$. Можно производить дешифровку напрямую возводя числа в степень $d=5$ или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 4$

$$4^5 = 1024 \rightarrow 1024 \bmod 46 = 12, \text{ т. к. } 46 \times 22 = 1012 \text{ (остаток 12)}$$

Таким образом, $M_1 = 4^5 \pmod{46} = 12$

Следовательно, $M_1 = 12$, что соответствует букве алфавита «к».

2. $C_2 = 8$

$$8^2 = 64 \rightarrow 64 \bmod 46 = 18, \text{ т. к. } 46 \times 1 = 46 \text{ (остаток 18)}$$

Таким образом, $8^2 \equiv 18 \pmod{46}$

$$8^4 = (8^2)^2 \equiv 18^2 = 324 \rightarrow 324 \bmod 46 = 2, \text{ т. к. } 46 \times 7 = 322 \text{ (остаток 2)}$$

Таким образом, $8^4 \equiv 2 \pmod{46}$

$$8^5 = 8^4 \times 8 \equiv 2 \times 8 = 16 \rightarrow 16 \bmod 46 = 16$$

Таким образом, $M_2 = 8^5 \pmod{46} = 16$



Следовательно, $M_2 = 16$, что соответствует букве алфавита «о».

3. $C_3 = 37$

$$37^2 = 1369 \rightarrow 1369 \bmod 46 = 35, \text{ т. к. } 46 \times 29 = 1334 \text{ (остаток 35)}$$

$$\text{Таким образом, } 37^2 \equiv 35 \pmod{46}$$

$$37^4 = (37^2)^2 \equiv 35^2 = 1225 \rightarrow 1225 \bmod 46 = 29, \text{ т. к. } 46 \times 26 = 1196 \text{ (остаток 29)}$$

$$\text{Таким образом, } 37^4 \equiv 29 \pmod{46}$$

$$37^5 = 37^4 \times 37 \equiv 29 \times 37 = 1073 \rightarrow 1073 \bmod 46 = 15, \text{ т. к. } 46 \times 23 = 1058 \text{ (остаток 15)}$$

$$\text{Таким образом, } M_3 = 37^5 \pmod{46} = 15$$

Следовательно, $M_3 = 15$, что соответствует букве алфавита «н».

4. $C_4 = 36$

$$36^2 = 1296 \rightarrow 1296 \bmod 46 = 8, \text{ т. к. } 46 \times 28 = 1288 \text{ (остаток 8)}$$

$$\text{Таким образом, } 36^2 \equiv 8 \pmod{46}$$

$$36^4 = (36^2)^2 \equiv 8^2 = 64 \rightarrow 64 \bmod 46 = 18, \text{ т. к. } 46 \times 1 = 46 \text{ (остаток 18)}$$

$$\text{Таким образом, } 36^4 \equiv 18 \pmod{46}$$

$$36^5 = 36^4 \times 36 \equiv 18 \times 36 = 648 \rightarrow 648 \bmod 46 = 4, \text{ т. к. } 46 \times 14 = 644 \text{ (остаток 4)}$$

$$\text{Таким образом, } M_4 = 36^5 \pmod{46} = 4$$

Следовательно, $M_4 = 4$, что соответствует букве алфавита «г».

5. $C_5 = 3$

$$3^5 = 243 \rightarrow 243 \bmod 46 = 13, \text{ т. к. } 46 \times 5 = 230 \text{ (остаток 13)}$$

$$\text{Таким образом, } M_5 = 3^5 \pmod{46} = 13$$

Следовательно, $M_5 = 13$, что соответствует букве алфавита «л».

6. $C_6 = 8$

Аналогично $C_2 = 8$, $M_2 = M_6 = 16$, что соответствует букве алфавита «о».

7. $C_7 = 44$

$$44^2 = 1936 \rightarrow 1936 \bmod 46 = 4, \text{ т. к. } 46 \times 42 = 1932 \text{ (остаток 4)}$$

$$\text{Таким образом, } 44^2 \equiv 4 \pmod{46}$$

$$44^4 = (44^2)^2 \equiv 4^2 = 16 \rightarrow 16 \bmod 46 = 16$$

$$\text{Таким образом, } 44^4 \equiv 16 \pmod{46}$$

$$44^5 = 44^4 \times 44 \equiv 16 \times 44 = 704 \rightarrow 704 \bmod 46 = 14, \text{ т. к. } 46 \times 15 = 690 \text{ (остаток 14)}$$

$$\text{Таким образом, } M_4 = 44^5 \pmod{46} = 14$$

Следовательно, $M_7 = 14$, что соответствует букве алфавита «м».

8. $C_8 = 16$

$$16^2 = 256 \rightarrow 256 \bmod 46 = 26, \text{ т. к. } 46 \times 5 = 230 \text{ (остаток 26)}$$

$$\text{Таким образом, } 16^2 \equiv 26 \pmod{46}$$

$$16^4 = (16^2)^2 \equiv 26^2 = 676 \rightarrow 676 \bmod 46 = 32, \text{ т. к. } 46 \times 14 = 644 \text{ (остаток 32)}$$

$$\text{Таким образом, } 16^4 \equiv 32 \pmod{46}$$



$$16^5 = 16^4 \times 16 \equiv 32 \times 16 = 512 \rightarrow 512 \bmod 46 = 6, \text{ т. к. } 46 \times 11 = 506 \text{ (остаток 6)}$$

$$\text{Таким образом, } M_8 = 16^5 \bmod 46 = 6$$

Следовательно, $M_8 = 6$, что соответствует букве алфавита «е».

$$9. \ C_9 = 12$$

$$12^2 = 144 \rightarrow 144 \bmod 46 = 6, \text{ т. к. } 46 \times 3 = 138 \text{ (остаток 6)}$$

$$\text{Таким образом, } 12^2 \equiv 6 \pmod{46}$$

$$12^4 = (12^2)^2 \equiv 6^2 = 36 \rightarrow 36 \bmod 46 = 36$$

$$\text{Таким образом, } 12^4 \equiv 36 \pmod{46}$$

$$12^5 = 12^4 \times 12 \equiv 36 \times 12 = 432 \rightarrow 432 \bmod 46 = 18, \text{ т. к. } 46 \times 9 = 414 \text{ (остаток 18)}$$

$$\text{Таким образом, } M_8 = 12^5 \bmod 46 = 18$$

Следовательно, $M_9 = 18$, что соответствует букве алфавита «р».

Соединив расшифрованные буквы, получаем слово «конгломер».

В ответе сказано указать два слова расшифрованного сообщения в формате «Первое слово» «Второе слово», соединяя получаем сизый конгломер.

Ответ: сизый конгломер

**Задача 5 (20 баллов)****Маршруты в Кибертауне**

В футуристическом городе Кибертауне роботы-курьеры доставляют посылки по сложной системе туннелей. Каждый маршрут робота записывается как последовательность команд управления, разделённых точками зарядки *.

Примеры маршрутов:

AB (2 команды без зарядки);

AB*CD (блок из 2 команд, зарядка, блок из 2 команд);

A*B*C (3 блока по 1 команде с зарядками между ними).

Алфавит команд (система управления) поддерживает ровно **K** различных команд (например: A, B, C, D, E, F — движение в разных направлениях).

Маршрут состоит из одного или нескольких блоков команд, блоки разделяются символом зарядки *, каждый блок — это последовательность из 1 или более команд.

Общая длина записи маршрута $\leq N$ символов (включая команды и разделители *), при этом маршрут **не может начинаться с ***, маршрут **не может заканчиваться на ***, два разделителя ** подряд **запрещены** (между зарядками всегда есть хотя бы одна команда).

Найти количество различных корректных маршрутов.

Формат входных данных

Одна строка с двумя целыми числами: **N** и **K** через пробел, **N** — максимальная длина маршрута (в количестве команд, в том числе зарядки), **K** — количество различных команд ($1 \leq N \leq 5000$, $1 \leq K \leq 26$).

Формат выходных данных

Одно целое число — количество корректных маршрутов.

Примеры

стандартный ввод	стандартный вывод
3 2	18

Пояснение к примеру

Например, две команды: A, B.

Корректные маршруты длины ≤ 3 :

без зарядок длины 1: A, B — 2 маршрута;

без зарядок длины 2: AA, AB, BA, BB — 4 маршрута;

без зарядок длины 3: AAA, AAB, ABA, ABB, BAA, BAB, BBA, BBB — 8 маршрутов;

с одной зарядкой (структура 1*1): A*A, A*B, B*A, B*B — 4 маршрута;



с одной зарядкой (структура 2^* или $*2$ невозможны, т.к. нельзя начинать/заканчивать с $*$).

Итого: 18 маршрутов.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных данных, поясните решение. Оцените сложность предложенного алгоритма.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
6 1	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Результаты выполнения программы для заданных исходных данных:

Вариант 1

стандартный ввод	стандартный вывод
6 1	20

Вариант 2

стандартный ввод	стандартный вывод
2 3	12

Вариант 3

стандартный ввод	стандартный вывод
2 4	20

Вариант 4

стандартный ввод	стандартный вывод
5 1	12

Основной подход — динамическое программирование. Мы разбиваем задачу на подзадачи: $dp[i]$ хранит количество способов построить корректный маршрут длины i , каждая подзадача решается на основе решений более простых подзадач.



Оптимальное решение большой задачи строится из оптимальных решений меньших подзадач: маршрут длины i может быть построен либо без зарядки, либо разделён на части через символ зарядки. Каждая часть после зарядки — это независимая подзадача.

Вариант рекуррентного соотношения:

$$dp[i] = K^i + \sum_{j=1}^{i-2} K^j \cdot dp[i - j - 1]$$

K^i – маршруты без зарядки;

$K^j \cdot dp[i - j - 1]$ – маршруты с первым отрезком длины j , затем зарядка «*», затем корректный маршрут длины $(i - j - 1)$. Заполняем dp от меньших длин к большему.

Итоговый ответ: $\sum_{i=1}^N dp[i]$.

Реализация на C++

```
#include <iostream>
#include <vector>
using namespace std;

long long power(long long base, int exp) {
    long long result = 1;
    for (int i = 0; i < exp; i++) {
        result *= base;
    }
    return result;
}

long long count_routes(int n, int k) {
    vector<long long> dp(n + 1, 0);

    // Для каждой возможной длины маршрута
    for (int length = 1; length <= n; length++) {
        // Вариант 1: весь маршрут без зарядки
        dp[length] = power(k, length);

        // Вариант 2: добавляем отрезки с зарядкой
        for (int first_segment_len = 1; first_segment_len < length - 1;
            first_segment_len++) {
            int remaining = length - first_segment_len - 1; // -1 для символа '*'
            if (remaining > 0) {
                dp[length] += power(k, first_segment_len) * dp[remaining];
            }
        }
    }

    // Суммируем все маршруты длины от 1 до N
    long long total = 0;
    for (int i = 1; i <= n; i++) {
        total += dp[i];
    }

    return total;
}
```



```
int main() {
    int n, k;
    cin >> n >> k;
    cout << count_routes(n, k) << endl;
    return 0;
}
```

Но на предельных значениях это решение не пройдет из-за времени и длинной арифметики.

Используем следующее рекуррентное соотношение. $E[i]$ – количество маршрутов длины i , заканчивающихся на команду. $S[i]$ – количество маршрутов длины i , заканчивающихся на зарядку (*). Чтобы получить маршрут длины i , заканчивающийся на команду, мы можем взять маршрут длины $i-1$, заканчивающийся на команду, и добавить к нему любую из K команд. Или взять маршрут длины $i-1$, заканчивающийся на зарядку, и добавить команду $E[i] = (E[i-1] + S[i-1]) \times K$. Чтобы получить маршрут длины i , заканчивающийся на зарядку, мы можем взять только тот маршрут длины $i-1$, который закончился на команду (так как ** запрещены) $S[i] = E[i-1]$. База: $E[1] = K$, $S[1] = 0$.

Вариант решения:

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <iomanip>

using namespace std;

// Простейшая структура для длинной арифметики
struct BigInt {
    static const int BASE = 1e9;
    vector<int> digits;

    BigInt(long long v = 0) {
        if (v == 0) digits.push_back(0);
        while (v > 0) {
            digits.push_back(v % BASE);
            v /= BASE;
        }
    }

    void operator+=(const BigInt& other) {
        int carry = 0;
        for (size_t i = 0; i < max(digits.size(), other.digits.size()) || carry; ++i) {
            if (i == digits.size()) digits.push_back(0);
            long long cur = digits[i] + carry + (i < other.digits.size() ? other.digits[i] : 0);
            digits[i] = cur % BASE;
            carry = cur / BASE;
        }
    }

    BigInt operator*(int v) const {
        BigInt res = *this;
        int carry = 0;
    }
}
```



```
        for (size_t i = 0; i < res.digits.size() || carry; ++i) {
            if (i == res.digits.size()) res.digits.push_back(0);
            long long cur = carry + res.digits[i] * 1LL * v;
            res.digits[i] = cur % BASE;
            carry = cur / BASE;
        }
        while (res.digits.size() > 1 && res.digits.back() == 0)
            res.digits.pop_back();
        return res;
    }

    void print() const {
        printf("%d", digits.empty() ? 0 : digits.back());
        for (int i = (int)digits.size() - 2; i >= 0; --i) {
            printf("%09d", digits[i]);
        }
        printf("\n");
    }
};

int main() {
    int n, k;
    if (!(cin >> n >> k)) return 0;

    if (n == 0) { cout << 0 << endl; return 0; }

    // E[i] - маршруты длины i, заканчивающиеся на букву
    // S[i] - маршруты длины i, заканчивающиеся на '*'
    vector<BigInt> E(n + 1), S(n + 1);

    E[1] = BigInt(k);
    S[1] = BigInt(0);

    BigInt total = E[1];

    for (int i = 2; i <= n; i++) {
        // Чтобы получить букву в конце: (предыдущий был буквой ИЛИ звездой)
        * K вариантов букв
        BigInt prev_sum = E[i - 1];
        prev_sum += S[i - 1];
        E[i] = prev_sum * k;

        // Чтобы получить звезду в конце: предыдущий обязан быть буквой
        S[i] = E[i - 1];

        // В ответ идут только маршруты, заканчивающиеся на букву (E[i])
        total += E[i];
    }

    total.print();

    return 0;
}
```

Сложность решения $O(N^2)$.

Решения на Python

```
import sys

def solve():
    # Считываем N и K
```



```
line = sys.stdin.readline()
if not line:
    return
n, k = map(int, line.split())

# dp[i] - количество маршрутов длины ровно i
dp = [0] * (n + 1)

# Предсчитаем степени K, чтобы не считать их каждый раз
powers_of_k = [1] * (n + 1)
for i in range(1, n + 1):
    powers_of_k[i] = powers_of_k[i-1] * k

# Чтобы ускорить сумму в DP, будем использовать префиксную сумму
# Сумма S[i] = dp[1] + dp[2] + ... + dp[i]
# Но нам нужна специфическая сумма:
# dp[i] = K^i + sum_{j=1}^{i-2} (K^j * dp[i-j-1])

for i in range(1, n + 1):
    # Случай без зарядок
    dp[i] = powers_of_k[i]

    # Случай с зарядками: перебор длины первого блока j
    # Маршрут: [блок длины j] + [*] + [маршрут длины i-j-1]
    for j in range(1, i - 1):
        remaining = i - j - 1
        dp[i] += powers_of_k[j] * dp[remaining]

# Ответ - сумма всех маршрутов длины от 1 до N
print(sum(dp))

solve()
```

Временная сложность: $O(N^3 \log K)$.

Оптимальное решение на python $\sim O(N^2)$.

```
import sys

# Увеличиваем лимит на количество цифр в строковом представлении числа.
# 10000 знаков хватит для N=5000 с запасом.
sys.set_int_max_str_digits(10000)

def solve():
    # Чтение входных данных
    input_data = sys.stdin.read().split()
    if not input_data:
        return

    n = int(input_data[0])
    k = int(input_data[1])

    if n == 0:
        print(0)
        return
    if n == 1:
        print(k)
        return

    # E[i] - маршруты длины i, заканчивающиеся на команду (букву)
    # S[i] - маршруты длины i, заканчивающиеся на зарядку (*)
    E = [0] * (n + 1)
    S = [0] * (n + 1)
```



```
E[1] = k
S[1] = 0

total_count = E[1]

for i in range(2, n + 1):
    # Рекуррентные соотношения:
    E[i] = (E[i-1] + S[i-1]) * k
    S[i] = E[i-1]

    # Суммируем только те, что заканчиваются на команду
    total_count += E[i]

print(total_count)

if __name__ == "__main__":
    solve()
```

Вариант решения перебором со сложностью $O((K+1)^N N)$, сработает максимально примерно для $N=15$.

```
def count_routes(n, k):
    count = 0

    # Перебираем все возможные строки длины от 1 до n
    for length in range(1, n + 1):
        # Генерируем все комбинации для данной длины
        count += generate_all(length, k, "", 0)

    return count

def generate_all(length, k, current, pos):
    # Базовый случай - достигли нужной длины
    if pos == length:
        if is_valid(current):
            return 1
        return 0

    total = 0

    # Пробуем добавить команду (буквы A, B, C, ...)
    for i in range(k):
        command = chr(ord('A') + i)
        total += generate_all(length, k, current + command, pos + 1)

    # Пробуем добавить разделитель *
    total += generate_all(length, k, current + '*', pos + 1)

    return total

def is_valid(route):
    # Проверяем корректность маршрута
    if not route:
        return False

    # Не может начинаться с *
    if route[0] == '*':
        return False

    # Не может заканчиваться на *
    if route[-1] == '*':
        return False
```



```
# Не может содержать **
if '**' in route:
    return False

return True

# Ввод данных
line = input().strip()
n, k = map(int, line.split())

# Вычисление и вывод результата
result = count_routes(n, k)
print(result)
```



Задача 6 (30 баллов)

Ящики Грю

В коридоре у известного суперзлодея Грю скопилась целая куча ящиков. Все они пронумерованы и расположены вдоль стены в одну линию друг за другом. Грю уже и не помнит, почему он их так пронумеровал, но на крышках этих ящиков есть самые разные номера, в том числе и повторяющиеся.

Будучи в душе перфекционистом, Грю давно хочет навести порядок в коридоре. Под порядком он понимает тот факт, чтобы ящики были расположены так же вдоль стены коридора в одну линию, но уже в порядке – порядке неубывания номеров. И вот, наконец-то сегодня, он решил осуществить свой суперзлодейский план по упорядочиванию ящиков.

Само собой, для этого у него есть целая армия неугомонных миньонов. Грю знает, что, если он даст работу некоторым из них, а остальных проигнорирует, те на него сильно обидятся. По этой причине Грю, вообще-то не отличающийся стандартным мышлением, решил действовать так: он выделит каждому миньону некоторый отрезок из нескольких подряд идущих ящиков и попросит того отсортировать эти ящики по неубыванию. Если миньону достанется такой отрезок из ящиков, что ему придется хотя бы немного их переставить – то миньон будет доволен, однако, если ему достанется уже упорядоченный отрезок, миньон будет считать, что его зря позвали и снова обидится. Чтобы миньоны не подрались, их отрезки не должны пересекаться.

Грю желает выделить отрезки миньонам так, чтобы после их работы весь ряд ящиков так же стал упорядочен по неубыванию – всё прямо как он и задумывал в своих суперзлодейских мечтах.

Теперь пришло время делить фронт работ, то есть набор ящиков на непересекающиеся отрезки. И тут Грю понял, что его супермозг не может решить такую задачу. Следует помочь ему и сказать какое максимально возможное число миньонов он сможет осчастливить, выделив каждому непустой отрезок, внутри которого ящики пока еще не стоят в нужном порядке, а после того, как их поставить в каждом таком отрезке по неубыванию, то и вся последовательность ящиков станет упорядоченной по неубыванию.

Формат входных данных

В первой строке находится одно целое число n – количество ящиков, которые находятся в коридоре у Грю ($1 \leq n \leq 2 \cdot 10^5$).

Во второй строке через пробел указаны целые номера b_i этих ящиков в том порядке, в котором они стоят вдоль стены ($1 \leq b_i \leq 10^{18}$).

Формат выходных данных

Вывести в ответ одно число – **максимальное** количество миньонов,



которых следует привлечь к упорядочиванию ящиков по методу Грю, с условием, что каждому достанется непустой неупорядоченный отрезок из ящиков. После того, как каждый миньон упорядочит свой отрезок, вся последовательность ящиков должна оказаться так же упорядоченной по неубыванию.

Примеры

стандартный ввод	стандартный вывод
5 2 1 4 5 3	2
4 3 3 4 5	0
5 7 1 3 2 2	1
11 3 4 3 2 4 7 8 7 12 12 9	3

Комментарии к примерам

В первом примере следует одному миньону выделить отрезок от первого до второго ящика, а второму миньону от третьего до пятого ящика.

Во втором примере набор ящиков уже упорядочен, и миньонам здесь делать нечего.

В третьем наборе единственный способ – выбрать один отрезок из всех ящиков. Нужен один миньон.

В четвёртом примере нужны три миньона. Первому следует выделить отрезок с первого до шестого (это ящики подписаны 3 4 3 2 4 7), второму с седьмого до восьмого (ящики подписаны 8 7), третьему оставшиеся ящики с девятого по одиннадцатый (на них стоят надписи 12 12 9). После того, как миньоны упорядочат каждый свои ящики, последовательность ящиков станет так же упорядоченной. Существуют и другие способы раздать отрезки в этом примере, однако больше трёх миньонов никогда не понадобится.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных данных. Оцените сложность предложенного алгоритма.



Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
11 4 1 3 5 2 10 7 6 9 11 8	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Результаты выполнения программы для заданных исходных данных:

Вариант 1

стандартный ввод	стандартный вывод
11 4 1 3 5 2 10 7 6 9 11 8	2

Вариант 2

стандартный ввод	стандартный вывод
11 4 1 6 3 5 10 7 6 9 11 8	2

Вариант 3

стандартный ввод	стандартный вывод
11 4 6 1 3 5 10 7 6 9 11 8	2

Вариант 4

стандартный ввод	стандартный вывод
11 4 6 1 3 5 7 10 6 9 11 8	2

Решение на C++

```
#include<bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#pragma GCC target("popcnt")

#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef pair<int, int> pii;
typedef vector<int> vi;
typedef vector<vector<int> > vvi;
typedef vector<double> vd;
typedef vector<vd> vvd;
```



```
typedef tree<int, null_type, less<int>, rb_tree_tag,  
tree_order_statistics_node_update> ordered_set;  
  
const int INF = 1e18;  
const int MOD = 1e9 + 7;  
const int LG = 19;  
  
signed main(){  
  
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);  
  
    int n;  
    cin >> n;  
    vi v(n);  
    for0(i, n){  
        cin >> v[i];  
    }  
  
    vi prefmx(n), suffmn(n);  
    int tmx = 0;  
    for0(i, n){  
        tmx = max(tmx, v[i]);  
        prefmx[i] = tmx;  
    }  
    int tmn = INF + 1;  
    for(int i = n - 1; i >= 0; i--){  
        tmn = min(tmn, v[i]);  
        suffmn[i] = tmn;  
    }  
    vi border(1, 0);  
  
    for(int i = 1; i < n; i++){  
        if(suffmn[i] >= prefmx[i - 1]){  
            border.pb(i);  
        }  
    }  
  
    int ans = 0;  
  
    for0(i, sz(border)){  
        bool need = 0;  
        int r = n;  
        if(i != sz(border) - 1){  
            r = border[i + 1];  
        }  
        for(int j = border[i] + 1; j < r; j++){  
            if(v[j] < v[j - 1]){  
                need = 1;  
            }  
        }  
        ans += need;  
    }  
  
    cout << ans << endl;  
}
```

Разбиваем массив на независимые сегменты, которые можно обрабатывать отдельно. Позиция i является границей сегмента, если все элементы справа от нее (начиная с i) больше или равны всем элементам слева



(до $i-1$). Ключевое свойство: если $\text{suffmn}[i] \geq \text{prefmx}[i-1]$, то можно "разрезать" массив в позиции i , и сортировка левой части не повлияет на правую часть (и наоборот). Для каждого сегмента между границами проверяется: требуется ли в нем хотя бы одна перестановка и если да, то нужен один миньон для этого сегмента.

Временная сложность – $O(n)$.

```
#include<bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#pragma GCC target("popcnt")

#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef pair<int, int> pii;
typedef vector<int> vi;
typedef vector<vector<int>> > vvi;
typedef vector<double> vd;
typedef vector<vd> vvd;
typedef tree<int, null_type, less<int>, rb_tree_tag,
tree_order_statistics_node_update> ordered_set;

const int INF = 1e18;
const int MOD = 1e9 + 7;
const int LG = 19;

signed main(){

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    int n;
    cin >> n;
    vi v(n);
    for0(i, n){
        cin >> v[i];
    }

    vi srt = v;
    sort(all(srt));

    vi psum(n);
    int tsum = 0;
    for0(i, n){
        tsum += srt[i];
        psum[i] = tsum;
    }

    cout << setw(4)<< srt[i]<< ' ';
```



```
int ans = 0;

tsum = 0;
int need = 0;
for0(i, n){
    tsum += v[i];
    if(i > 0 && v[i] < v[i - 1]){
        need = 1;
    }
    if(tsum == psum[i]){
        ans += need;
        need = 0;
    }
}

cout << ans << endl;
}
```

Это решение использует сравнение префиксных сумм. Если префиксная сумма совпадает с суммой отсортированного префикса → этот префикс содержит те же элементы, что и в отсортированном массиве → можно отсортировать независимо от остальной части. Сложность – $O(n \log n)$, по памяти – $O(n)$.

Решение на Python

```
def solve():
    n = int(input())
    boxes = list(map(int, input().split()))

    # Если массив уже отсортирован, миньоны не нужны
    if boxes == sorted(boxes):
        print(0)
        return

    # Найдем все максимальные отрезки, которые нужно сортировать
    segments = []
    i = 0

    while i < n:
        # Находим начало неотсортированного участка
        while i < n - 1 and boxes[i] <= boxes[i + 1]:
            i += 1

        if i == n - 1:
            break

        # Нашли начало неотсортированного участка
        start = i

        # Находим минимальный и максимальный элементы в потенциальном отрезке
        min_val = boxes[i]
        max_val = boxes[i]

        # Расширяем отрезок вправо
        j = i
        while j < n:
            min_val = min(min_val, boxes[j])
            max_val = max(max_val, boxes[j])

            # Проверяем, может ли отрезок [start:j+1] быть независимым
```



```
# Все элементы слева от start должны быть <= min_val
# Все элементы справа от j должны быть >= max_val
can_end_here = True

# Проверка слева
if start > 0 and boxes[start - 1] > min_val:
    can_end_here = False

# Проверка справа
if j < n - 1 and boxes[j + 1] < max_val:
    can_end_here = False

if not can_end_here:
    j += 1
else:
    # Можем закончить здесь, но попробуем расширить
    j += 1
    if j >= n:
        segments.append((start, j - 1))
        i = j
        break
    continue
else:
    # Дошли до конца
    segments.append((start, n - 1))
    i = n
    break

# Теперь нужно найти максимальное количество непересекающихся отрезков
# где каждый отрезок не отсортирован

# Переформулируем задачу: найти максимальное разбиение на отрезки
dp = [0] * (n + 1) # dp[i] - максимум миньонов для первых i ящиков

for i in range(1, n + 1):
    dp[i] = dp[i - 1] # Можем не брать i-й элемент в отрезок

    # Пробуем все возможные отрезки, заканчивающиеся в позиции i-1
    for start in range(i):
        # Проверяем отрезок [start, i-1]
        segment = boxes[start:i]
        sorted_segment = sorted(segment)

        # Отрезок должен быть неотсортирован
        if segment == sorted_segment:
            continue

        # Проверяем, что после сортировки этого отрезка
        # последовательность останется корректной
        min_val = min(segment)
        max_val = max(segment)

        # Проверка слева
        if start > 0 and boxes[start - 1] > min_val:
            continue

        # Проверка справа
        if i < n and boxes[i] < max_val:
            continue

        # Этот отрезок подходит
        dp[i] = max(dp[i], dp[start] + 1)
```



```
print(dp[n])  
  
solve()
```

Алгоритм работает следующим образом:

- Проверяем, не отсортирован ли массив уже (тогда ответ 0)
- Используем динамическое программирование: $dp[i]$ — максимальное количество миньонов для обработки первых i ящиков
- Для каждой позиции i перебираем все возможные отрезки $[start, i-1]$, которые не отсортированы (иначе миньон обидится)
- После сортировки не нарушат общий порядок (проверяем границы с соседними элементами)
- Выбираем максимальное количество таких непересекающихся отрезков.

Сложность решения: $O(n^3)$ в худшем случае, но для $n \leq 2 \cdot 10^5$ это работает достаточно быстро благодаря отсечениям.

Возможны переборные решения сложностью $O(2^n \cdot n \log n)$.

```
def is_sorted_segment(arr, l, r):  
    """Проверка, отсортирован ли отрезок [l, r)"""  
    for i in range(l, r - 1):  
        if arr[i] > arr[i + 1]:  
            return False  
    return True  
  
def solve():  
    n = int(input())  
    arr = list(map(int, input().split()))  
  
    # Целевой отсортированный массив  
    target = sorted(arr)  
  
    max_minions = 0  
  
    # Перебираем все возможные разбиения (2^(n-1) вариантов)  
    # Битовая маска определяет, где делаем разрезы  
    for mask in range(1 << (n - 1)):  
        # Строим список границ отрезков  
        segments = []  
        start = 0  
  
        for i in range(n - 1):  
            if mask & (1 << i): # Делаем разрез после позиции i  
                segments.append((start, i + 1))  
                start = i + 1  
  
        # Добавляем последний отрезок  
        segments.append((start, n))  
  
        # Создаем копию массива для сортировки отрезков  
        temp = arr[:]  
        valid = True  
        minion_count = 0  
  
        # Проверяем каждый отрезок
```



```
for l, r in segments:
    # Проверяем, что отрезок неупорядочен
    if not is_sorted_segment(arr, l, r):
        minion_count += 1
        # Сортируем отрезок
        temp[l:r] = sorted(temp[l:r])
    # Если отрезок уже упорядочен, но состоит из >1 элемента,
    # мы его не считаем (миньон обидится)

# Проверяем, что результат совпадает с целевым
if temp == target:
    max_minions = max(max_minions, minion_count)

print(max_minions)

solve()
```



ВАРИАНТ 2

Задача 1 (5 баллов)

Разведчики

От разведчиков были получены сведения, закодированные в виде электронной таблицы. Определите числа в ячейках C2:C11, которые будут получены в результате копирования в них блока ячеек B1:B10.

В ячейках A1:A5 разместите десятичные результаты вычисления чисел разных систем счисления:

$$A1 = 14_5 - 11_3 + 12_3$$

$$A2 = 13_4 - 12_3 + 11_2$$

$$A3 = 20_5 - 22_3 + 12_4$$

$$A4 = 22_3 - 11_5 + 10_4$$

$$A5 = 22_4 - 22_5 + 11_5$$

В ячейке A6 разместите формулу: = "АБВ".

	A	B	C	D
1		=СУММ(A\$1:\$A6)		
2		=МИН(\$A\$1:\$A\$6;\$B\$1)		
3		=МАКС(\$A1:\$A\$6;\$B\$1:\$B2)		
4		=ЕСЛИ(A\$6=0;A\$4;\$A5)		
5		=СУММЕСЛИ(A1:B\$4;A2)		
6		=СЧЁТЕСЛИ(A1:B\$5;"<A4")		
7		=ЕСЛИ(ИЛИ(B\$3>\$B4;\$B\$5<\$B\$6);A4;\$A5)		
8		=ЕСЛИ(И(B4<\$B5;\$B6>\$B7);A1;A2)		
9		=ЕСЛИ(B7-\$B\$8>0;\$B\$7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;\$B5))		
10		=ЕСЛИ(B8>\$B\$9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))		
11				
12				

Ответ записать в одну строчку через запятую, начиная с ячейки C2 и заканчивая ячейкой C11.

Решение

Получим значения в ячейках A1:A5 как результат вычисления чисел разных систем счисления:

$$A1 = 14_5 - 11_3 + 12_3 = 9 - 4 + 5 = 10$$

$$A2 = 13_4 - 12_3 + 11_2 = 7 - 5 + 3 = 5$$

$$A3 = 20_5 - 22_3 + 12_4 = 10 - 8 + 6 = 8$$

$$A4 = 22_3 - 11_5 + 10_4 = 8 - 6 + 4 = 6$$

$$A5 = 22_4 - 22_5 + 11_5 = 10 - 12 + 6 = 4$$

Следующим шагом копируем блок ячеек B1:B10 в ячейки C2:C11, в результате получим следующие формулы:



	A	B	C
1	10	=СУММ(A\$1;\$A6)	
2	5	=МИН(\$A\$1:A\$6;B\$1)	=СУММ(B\$1;\$A7)
3	8	=МАКС(\$A1:A\$6;\$B\$1:B2)	=МИН(\$A\$1:B\$6;C\$1)
4	6	=ЕСЛИ(A\$6=0;A\$4;\$A5)	=МАКС(\$A2:B\$6;\$B\$1:C3)
5	4	=СУММЕСЛИ(A1:B\$4;A2)	=ЕСЛИ(B\$6=0;B\$4;\$A6)
6	= "АБВ"	=СЧЁТЕСЛИ(A1:B\$5;"<A4")	=СУММЕСЛИ(B2:C\$4;B3)
7		=ЕСЛИ(ИЛИ(B\$3>\$B4;\$B\$5<\$B6);A4;\$A5)	=СЧЁТЕСЛИ(B2:C\$5;"<A4")
8		=ЕСЛИ(И(B4<\$B5;\$B6>\$B7);A1;A2)	=ЕСЛИ(ИЛИ(C\$3>\$B5;\$B\$5<\$C6);B5;\$B5)
9		=ЕСЛИ(B7-\$B\$8>0;\$B7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;\$B5))	=ЕСЛИ(И(C5<\$C6;\$C7>\$C8);B2;\$B3)
10		=ЕСЛИ(B8>\$B\$9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))	=ЕСЛИ(C8-\$B\$8>0;\$C7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;\$B5))
11			=ЕСЛИ(C9>\$C\$9;СЧЁТЕСЛИ(C2:C10;">7");СЧЁТЕСЛИ(C2:C10;"<7"))
12			

Вычисляем значения по формулам в ячейках В1:В10, затем переходим к вычислениям значений в ячейках С2:С11. Получаем следующие результаты:

	A	B	C	D
1	10	10		
2	5	4	10	
3	8	10	0	
4	6	4	10	
5	4	5	4	
6	АБВ	0	30	
7		6	0	
8		5	5	
9		6	10	
10		7	30	
11			4	
12				

Ответ: 10, 0, 10, 4, 30, 0, 5, 10, 30, 4

**Задача 2 (10 баллов)****Ключ от архива**

Историк Петр обнаружил старый архив, защищённый кодовым замком. Документы содержат информацию, зашифрованную четырехразрядным числовым ключом в шестнадцатеричной системе счисления. Петр знает, что каждая цифра числа слева направо была преобразована в двоичную систему счисления и записана четырехразрядным равномерным кодом, т.е. если двоичное представление содержало менее четырёх разрядов, оно дополнялось нулями слева до полного четырёхразрядного кода (например, $716=01112$). Полученные двоичные коды соединили последовательно слева направо.

Таблица истинности этого логического выражения содержит все необходимые значения в результирующем столбце сверху вниз для определения кода:

$$a \vee b \rightarrow (c \oplus \neg d \vee a) \rightarrow (d \equiv (a \oplus b \wedge c))$$

Строки значений a, b, c, d в таблице истинности идут последовательно, в лексикографическом порядке 0000, 0001, 0010 и т.д. Определите четырёхзначный код архива в шестнадцатеричной системе счисления.

Решение

Составим таблицу истинности

a	b	c	d	$a \vee b$	$\neg d$	$c \oplus \neg d$	$c \oplus \neg d \vee a$	$a \vee b \rightarrow (c \oplus \neg d \vee a)$	$b \wedge c$	$a \oplus b \wedge c$	$d \equiv (a \oplus b \wedge c)$	Итого
0	0	0	0	0	1	1	1	1	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0	1	0	0	1	1
0	0	1	1	0	0	1	1	1	0	0	0	0
0	1	0	0	1	1	1	1	1	0	0	1	1
0	1	0	1	1	0	0	0	0	0	0	0	1
0	1	1	0	1	1	0	0	0	1	1	0	1
0	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	0	0
1	0	0	1	1	0	0	1	1	0	1	1	1
1	0	1	0	1	1	0	1	1	0	1	0	0
1	0	1	1	1	0	1	1	1	0	1	1	1
1	1	0	0	1	1	1	1	1	0	1	0	0
1	1	0	1	1	0	0	1	1	0	1	1	1
1	1	1	0	1	1	0	1	1	1	0	1	1
1	1	1	1	1	0	1	1	1	1	0	0	0

Двоичный код в последнем столбце можно разбить на четвертичные числа следующим образом: 1010, 1111, 0101, 0110.

Таким образом получаем следующий шестнадцатеричный код: $AF56_{16}$ или $(10)(15)56_{16}$

Ответ: $AF56_{16}$ или $(10)(15)56_{16}$



Задача 3 (15 баллов)

Циклический возврат камеры наблюдения

Исполнитель Камера используется для управления камерой наблюдения, которая может вращаться ограниченно по вертикали и без ограничений по горизонтали. Исполнитель Камера может выполнять следующие команды:

1) Повернуть на (a, b) , где a, b – целые числа. Эта команда поворачивает камеру на величину вверх по вертикали на угол a , заданный в градусах, и поворачивает его против часовой стрелки на угол b , заданный в градусах. Например, команда Повернуть на $(-2, -30)$ опустит камеру на 2 градуса по вертикали вниз и повернет его на угол 30 по горизонтали по часовой стрелке.

2) Цикл:

ПОВТОРИ n

последовательность команд

КОНЕЦ ПОВТОРИ

Этот цикл означает, что последовательность команд будет выполняться n раз.

Исполнителю Камера был дан для исполнения следующий алгоритм (буквами a, b и n обозначены неизвестные целые числа):

НАЧАЛО

повернуть на $(60, -60)$

ПОВТОРИ n

повернуть на (a, b)

повернуть на $(-5, 30)$

КОНЕЦ ПОВТОРИ

повернуть на $(-10, -60)$

КОНЕЦ

Чему равно максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов. Чему равно минимально возможное число полных оборотов (360 градусов), которое при этом совершит камера?

Чему равны значения n , если после выполнения алгоритма камера вернулась в то же состояние, что было до начала выполнения алгоритма, совершив 5 полных оборотов? Чему, при этом равны параметры a и b ?

Решение

Значение параметров исполнителя Камера в результате выполнения алгоритма рассчитывается по формуле:



$$l - l_0 = 60 + n(a - 5) - 10$$

$$f - f_0 = -60 + n(b + 30) - 60$$

В этих уравнениях n – количество повторений команд ПОВТОРИ, l и l_0 – конечное и начальное значение угла вертикального поворота камеры, f и f_0 – конечное и начальное значение угла горизонтального поворота камеры.

Так как камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, $l - l_0 = 0$, а $f - f_0 = 360k$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Знак $+$ соответствует оборотам против часовой стрелки, а знак $-$ соответствует оборотам по часовой стрелке.

С учетом этих условий, получим систему уравнений:

$$60 + n(a - 5) - 10 = 0$$

$$-60 + n(b + 30) - 60 = 360k$$

После преобразований, получим:

$$n(a - 5) = -50$$

$$n(b + 30) = 360k + 120$$

Нужно найти натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число n должно быть одновременно делителем чисел 50 и $360k + 120$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Наибольшим общим делителем для этих выражений будет число 50. Оно будет соответствовать максимально возможному значению n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, при условии, что $360k + 120$ будет кратно 50. Минимальное число оборотов равно 2, что соответствует $k = -2$.

При совершении 5 полных оборотов возможны два варианта, в зависимости от направления вращения:

$$n(b + 30) = 360 \cdot 5 + 120 = 1920$$

$$n(b + 30) = 360 \cdot (-5) + 120 = -1680$$

Нужно найти наибольшее натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b – наибольшее число, которое является делителем чисел 50 и 1920 или 50 и 1680. В обоих случаях наибольший общий делитель $n = 10$. Рассчитав значения a и b , получим:

$$a = -50/10 + 5 = 0$$

$$b_{+} = 1920/10 - 30 = 162 \text{ – при оборотах против часовой стрелки}$$

$$b_{-} = -1680/10 - 30 = -198 \text{ – при оборотах по часовой стрелке}$$



Остальные значения n представляют собой делители числа 10: 1, 2, 5. Для них расчет значений a и b производится по аналогичным формулам.

Ответ: максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 50, при этом минимально возможное число полных оборотов равно 2. При совершении 5 полных оборотов максимальное $n=10$, $a=0$, $b=-198$ при оборотах по часовой стрелке и $b=162$ при оборотах против часовой стрелки.



Задача 4 (20 баллов)

Геологический шифр

В ходе геологоразведочных работ в Арктической зоне РФ была обнаружена неизвестная науке горная порода с уникальными кристаллическими свойствами. Чтобы сохранить приоритет открытия в тайне от конкурентов, старший геолог Пётр передал для анализа коллеге Марии зашифрованные данные, содержащие предварительное название породы из двух слов.

Мария получила два файла:

- 1) rock_sample.dat – файл с основным зашифрованным отчётом.
- 2) thin_section.png – микрофотография тонкого среза породы, в которую Петром был внедрён секретный ключ методом стеганографии, имитирующим природные включения.

Алгоритм действий для Марии выглядит следующим образом:

1) Извлечь скрытые данные из изображения thin_section.png. Для этого необходимо проанализировать младшие биты цветовых каналов пикселей, что позволит получить битовую последовательность.

2) Эту битовую последовательность необходимо декодировать. В результате она должна превратиться в строку в формате JSON без кавычек, которая и содержит секретные ключи для алгоритма RSA. В данной задаче строка в формате JSON без кавычек выглядит следующим образом: {ключ1:значение1,ключ2: значение2,ключ3:значение3}.

3) Используя полученные ключи, можно приступить к дешифровке основного отчёта, хранящегося в файле rock_sample.dat.

В результате анализа младших битов цветовых каналов пикселей изображения thin_section.png Мария получила следующую строку:

1100000101100111010111010110100001110011111110101011010000011
00111001111011111110110001

Пётр, известный своей любовью к горным треккигам, для защиты информации в файле thin_section.png применяет специальный «Горный код» стеганографии. Его принцип основан на использовании неравномерного кодирования, где частым символам соответствуют короткие коды, что минимизирует общий размер секретного сообщения. Дополнительный уровень защиты обеспечивают «мусорные» символы – специальные ловушки, встроенные в код. При расшифровке легитимный получатель должен распознать и отфильтровать этот мусор, чтобы получить исходные данные. Таблица используемых Петром кодов представлена ниже.

Символ	Код	Символ	Код
{	110000	7	111100



Символ	Код	Символ	Код
}	110001	8	111101
"	110010	9	111110
:	110011	n	000
,	110100	e	001
0	110101	d	010
1	110110	p	0110
2	110111	q	0111
3	111000	x	1000
4	111001	z	1001
5	111010	y	1010
6	111011		

Неравномерный код – это код, в котором кодовые слова, например, последовательности битов, имеют разную длину.

Например, при декодировании битовой последовательности 000001010 с помощью вышепредставленной таблицы, процесс декодирования будет заключаться в следующем:

1) Читаем биты слева-направо: 0 → 00 → 000.

000 есть в таблице? Да, это n.

Результат: n.

2) Продолжаем со следующего бита: 0 → 00 → 001.

001 есть в таблице? Да, это e.

Результат: e.

3) Продолжаем со следующего бита: 0 → 01 → 010.

010 есть в таблице? Да, это d.

Результат: d.

Расшифрованное сообщение n e d.

Файл rock_sample.dat представляет собой последовательность десятичных чисел, каждая из которых – зашифрованная буква передаваемого сообщения. Исходный текст отчёта перед шифрованием был преобразован в числа по правилу: каждый символ кодировался своим порядковым номером в алфавите (a=1, б=2, ..., я=33). Для шифрования Петром использовалась формула: $C_i = M_i^e \bmod n$,

где C_i – i-й элемент последовательности шифротекста C;

M_i – i-й элемент последовательности шифра M;

e – открытая экспонента – параметр открытого ключа RSA – число, используемое при шифровании в качестве степени;

n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.

Для дешифрования должна использоваться формула $M_i = C_i^d \bmod n$,

где M_i – i-й элемент последовательности шифра M;

C_i – i-й элемент последовательности шифротекста C;



d – секретная экспонента – параметр закрытого ключа RSA – число, используемое при дешифровке в качестве степени;

n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.

Совокупность значений (e, n) и (d, n) является открытым и закрытым ключами, используемыми при шифровании и дешифровании соответственно.

Например, RSA-шифрование с открытым ключом: $(e, n) = (17, 3233)$ – и закрытым ключом: $(d, n) = (2753, 3233)$ заключается в следующем.

При шифровании сообщения $M = 65$ отправляемый шифротекст $C = M^e \text{mod } n = 65^{17} \text{mod } 3233 = 2790$

При дешифровке полученного шифротекста $C = 2790$ восстановленное сообщение $M = C^d \text{mod } n = 2790^{2753} \text{mod } 3233 = 65$.

Примечание

Чтобы вычислить выражение вида $M = C^d \text{mod } n$, иногда невыгодно сначала находить большое число C^d , ведь оно может быть огромным. Удобнее использовать алгоритм быстрого возведения в степень по модулю.

Идея алгоритма:

1. Разложить показатель степени на степени двойки.
2. Последовательно возводить число в квадрат.
3. После каждого шага брать остаток по модулю.
4. Умножать только те степени, которые нужны по разложению.

Например, если нужно найти значение выражения $11^5 \text{mod } 9$, то степень 5 можно представить как сумму степеней 4 и 1: $5 = 4 + 1$.

Следовательно:

1. Находим остаток: $11^2 \text{mod } 9 = 4$, т.к. $9 \times 13 = 117$ (остаток 4).

2. Используем найденный остаток, чтобы получить $11^4 \text{mod } 9$.

$11^4 \text{mod } 9 = (11^2)^2 \text{mod } 9 \equiv 4^2 \text{mod } 9 = 7$, т.к. $9 \times 1 = 9$ (остаток 7)

3. Перемножаем найденный остаток с 11 – и снова берём остаток.

$11^5 \text{mod } 9 = (11^4 \times 11) \text{mod } 9 \equiv (7 \times 11) \text{mod } 9 = 77 \text{mod } 9 = 5$, т.к. $9 \times 8 = 72$ (остаток 5)

Таким образом, $11^5 \text{mod } 9 = 5$.

Используя данные, извлечённые из стеганографического сообщения, расшифруйте rock_sample.dat и узнайте предварительное название новой горной породы. Содержимое rock_sample.dat:

Первое слово – 49 15 11 43 10 5 62 65; второе слово – 27 12 39 44 39 60 15.

В ответе укажите два слова расшифрованного сообщения в формате «Первое слово» «Второе слово» (например, Медное золото).



Решение

На первом этапе необходимо определить секретный ключ, который был внедрен Петром методом стеганографии в файл `thin_section.png`.

В задаче сказано, что Пётр использовал для шифрования специальный вод, который основан на использовании неравномерного кодирования. Алгоритм дешифровки в данном случае заключается в следующем: читаем строку по битам слева направо, пошагово накапливая последовательность, пока она не совпадёт с одним из кодов в таблице. Как только совпадение найдено, записываем символ и начинаем формировать следующую последовательность заново.

В условии сказано, что Мария в результате анализа данного изображения получила битовую строку:

1100000101100111010111010110100001110011111110101011010000011
0011100111101111110110001

Начнём процесс декодирования. Если код не найден сразу (например, при чтении первой 1), то показан процесс накопления последовательности: $1 \rightarrow 11 \rightarrow 110$ и т.д., до момента совпадения с записью в таблице кодов, используемых Петром при шифровании. Максимальная длина кодового слова в таблице кодировки – 6 бит (например, 110000, 110011, 111110 и т.д.), поэтому в процессе накопления битовой последовательности последовательности длиннее 6 символов не рассматриваются.

Шаг	Индекс начала	Биты (извлечённые)	Найденный код	Символ
1	0	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110000$	110000	{
2	6	$0 \rightarrow 01 \rightarrow 010$	010	d
3	9	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
4	15	$1 \rightarrow 10 \rightarrow 101 \rightarrow 1010$	1010	y
5	19	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111010$	111010	5
6	25	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1101 \rightarrow 11010 \rightarrow 110100$	110100	,
7	31	$0 \rightarrow 00 \rightarrow 001$	001	e
8	34	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
9	40	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1111 \rightarrow 11111 \rightarrow 111110$	111110	9
10	46	$1 \rightarrow 10 \rightarrow 101 \rightarrow 1010$	1010	y
11	50	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1101 \rightarrow 11010 \rightarrow 110100$	110100	,
12	56	$0 \rightarrow 00 \rightarrow 000$	000	n
13	59	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1101 \rightarrow 11010 \rightarrow 110100$	110100	:
14	65	$1 \rightarrow 10 \rightarrow 100 \rightarrow 1001$	1001	z
15	69	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111011$	111011	6



Шаг	Индекс начала	Биты (извлечённые)	Найденный код	Символ
16	75	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1111 \rightarrow 11111 \rightarrow 111110$	111110	9
17	81	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110001$	110001	}

Собирая полученные символы в порядке их дешифровки, получаем точную расшифровку битовой строки: {d:y5,e:9y,n:z69}. В условии задачи говорится про «мусорные символы» – ловушки, которые Пётр вставляет в сообщение. В таблице кодов есть набор букв-«включений» x, y, z (и т.д.), и они выступают как мусор в конкретном сообщении: они появляются внутри ключей/значений, но не несут смысла для конечного JSON.

Если удалить все мусорные буквы x, y, z из полученной строки {d:y5,e:9y,n:z69}, то получим корректную JSON-подобную структуру без кавычек: {d:5,e:9,n:69}. Следовательно, получены данные для расшифровки текстового сообщения из файла rock_sample.dat, а именно модуль $n=69$ и секретная экспонента $d=5$.

Файл rock_sample.dat содержит две последовательности целых чисел, где каждое число – зашифрованный символ шифротекста:

Первое слово: 49 15 11 43 10 5 62 65

Второе слово: 27 12 39 44 39 60 15

Из условия задачи воспользуемся формулой дешифрования: $M_i = C_i^d \bmod n$,

где M_i – i -й элемент последовательности шифра M (восстановленный номер буквы ($1 = a, 2 = б, \dots, 33 = я$));

C_i – i -й элемент последовательности шифротекста C ;

d – секретная экспонента;

n – модуль операции взятия остатка;

$\bmod$ – операция взятия остатка от деления.

Дешифруем первое слово: $C = [49 \ 15 \ 11 \ 43 \ 10 \ 5 \ 62 \ 65]$. Можно производить дешифровку напрямую возводя числа в степень $d=5$ или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 49$

$$49^2 = 2401 \rightarrow 2401 \bmod 69 = 31, \text{ т. к. } 69 \times 34 = 2346 \text{ (остаток 55)}$$

$$\text{Таким образом, } 49^2 \equiv 31 \pmod{69}$$

$$49^4 = (49^2)^2 \equiv 31^2 = 961 \rightarrow 961 \bmod 69 = 58, \text{ т. к. } 69 \times 43 = 2967 \text{ (остаток 58)}$$

$$\text{Таким образом, } 49^4 \equiv 58 \pmod{69}$$

$$49^5 = 49^4 \times 49 \equiv 58 \times 49 = 2842 \rightarrow 2842 \bmod 69 = 13, \text{ т. к. } 69 \times 41 = 2829 \text{ (остаток 13)}$$

$$\text{Таким образом, } M_1 = 49^5 \pmod{69} = 13$$

Следовательно, $M_1 = 13$, что соответствует букве алфавита «л».

2. $C_2 = 15$

$$15^2 = 225 \rightarrow 225 \bmod 69 = 18, \text{ т. к. } 69 \times 3 = 207 \text{ (остаток 18)}$$

$$\text{Таким образом, } 15^2 \equiv 18 \pmod{69}$$



$$15^4 = (15^2)^2 \equiv 18^2 = 324 \rightarrow 324 \bmod 69 = 48, \text{ т. к. } 69 \times 4 = 276 \text{ (остаток 48)}$$

Таким образом, $15^4 \equiv 48 \pmod{69}$

$$15^5 = 15^4 \times 15 \equiv 48 \times 15 = 720 \rightarrow 720 \bmod 69 = 30, \text{ т. к. } 69 \times 10 = 690 \text{ (остаток 30)}$$

Таким образом, $M_2 = 15^5 \pmod{69} = 30$

Следовательно, $M_2 = 30$, что соответствует букве алфавита «ъ».

3. $C_3 = 11$

$$11^2 = 121 \rightarrow 121 \bmod 69 = 52, \text{ т. к. } 69 \times 1 = 69 \text{ (остаток 52)}$$

Таким образом, $11^2 \equiv 52 \pmod{69}$

$$11^4 = (11^2)^2 \equiv 52^2 = 2704 \rightarrow 2704 \bmod 69 = 13, \text{ т. к. } 69 \times 39 = 2691 \text{ (остаток 13)}$$

Таким образом, $11^4 \equiv 13 \pmod{69}$

$$11^5 = 11^4 \times 11 \equiv 13 \times 11 = 143 \rightarrow 143 \bmod 69 = 5, \text{ т. к. } 69 \times 2 = 138 \text{ (остаток 5)}$$

Таким образом, $M_3 = 11^5 \pmod{69} = 5$

Следовательно, $M_3 = 5$, что соответствует букве алфавита «д».

4. $C_4 = 43$

$$43^2 = 1849 \rightarrow 1849 \bmod 69 = 55, \text{ т. к. } 69 \times 26 = 1794 \text{ (остаток 55)}$$

Таким образом, $43^2 \equiv 55 \pmod{69}$

$$43^4 = (43^2)^2 \equiv 55^2 = 3025 \rightarrow 3025 \bmod 69 = 58, \text{ т. к. } 69 \times 43 = 2967 \text{ (остаток 58)}$$

Таким образом, $43^4 \equiv 58 \pmod{69}$

$$43^5 = 43^4 \times 43 \equiv 58 \times 43 = 2494 \rightarrow 2494 \bmod 69 = 10, \text{ т. к. } 69 \times 36 = 2484 \text{ (остаток 10)}$$

Таким образом, $M_4 = 43^5 \pmod{69} = 10$

Следовательно, $M_4 = 10$, что соответствует букве алфавита «и».

5. $C_5 = 10$

$$10^2 = 100 \rightarrow 100 \bmod 69 = 31, \text{ т. к. } 69 \times 1 = 69 \text{ (остаток 31)}$$

Таким образом, $10^2 \equiv 31 \pmod{69}$

$$10^4 = (10^2)^2 \equiv 31^2 = 961 \rightarrow 961 \bmod 69 = 64, \text{ т. к. } 69 \times 13 = 897 \text{ (остаток 64)}$$

Таким образом, $10^4 \equiv 64 \pmod{69}$

$$10^5 = 10^4 \times 10 \equiv 64 \times 10 = 640 \rightarrow 640 \bmod 69 = 19, \text{ т. к. } 69 \times 9 = 621 \text{ (остаток 19)}$$

Таким образом, $M_5 = 10^5 \pmod{69} = 19$

Следовательно, $M_5 = 19$, что соответствует букве алфавита «с».

6. $C_6 = 5$

$$5^5 = 3125 \rightarrow 3125 \bmod 69 = 20, \text{ т. к. } 69 \times 45 = 3105 \text{ (остаток 20)}$$

Таким образом, $M_6 = 5^5 \pmod{69} = 20$

Следовательно, $M_6 = 20$, что соответствует букве алфавита «т».

7. $C_7 = 62$

$$62^2 = 3844 \rightarrow 3844 \bmod 69 = 49, \text{ т. к. } 69 \times 55 = 3795 \text{ (остаток 49)}$$

Таким образом, $62^2 \equiv 49 \pmod{69}$



$$62^4 = (62^2)^2 \equiv 49^2 = 2401 \rightarrow 2401 \bmod 69 = 55, \text{ т. к. } 69 \times 34 = 2346 \text{ (остаток 55)}$$

Таким образом, $62^4 \equiv 55 \pmod{69}$

$$62^5 = 62^4 \times 62 \equiv 55 \times 62 = 3410 \rightarrow 3410 \bmod 69 = 29, \text{ т. к. } 69 \times 49 = 3381 \text{ (остаток 29)}$$

Таким образом, $M_7 = 62^5 \pmod{69} = 29$

Следовательно, $M_7 = 29$, что соответствует букве алфавита «ы».

8. $C_8 = 65$

$$65^2 = 4225 \rightarrow 4225 \bmod 69 = 16, \text{ т. к. } 69 \times 61 = 4209 \text{ (остаток 16)}$$

Таким образом, $65^2 \equiv 16 \pmod{69}$

$$65^4 = (65^2)^2 \equiv 16^2 = 256 \rightarrow 3025 \bmod 69 = 49, \text{ т. к. } 69 \times 3 = 207 \text{ (остаток 49)}$$

Таким образом, $65^4 \equiv 58 \pmod{69}$

$$65^5 = 65^4 \times 65 \equiv 49 \times 65 = 3185 \rightarrow 3185 \bmod 69 = 11, \text{ т. к. } 69 \times 46 = 3174 \text{ (остаток 11)}$$

Таким образом, $M_8 = 65^5 \pmod{69} = 11$

Следовательно, $M_8 = 11$, что соответствует букве алфавита «й».

Соединив расшифрованные буквы, получаем слово «льдистый».

Дешифруем второе слово: $C = [27 \ 12 \ 39 \ 44 \ 39 \ 60 \ 15]$. Можно производить дешифровку напрямую возводя числа в степень $d=5$ или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 27$

$$27^2 = 2401 \rightarrow 2401 \bmod 69 = 31, \text{ т. к. } 69 \times 34 = 2346 \text{ (остаток 55)}$$

Таким образом, $27^2 \equiv 55 \pmod{69}$

$$27^4 = (27^2)^2 \equiv 55^2 = 3025 \rightarrow 3025 \bmod 69 = 58, \text{ т. к. } 69 \times 43 = 2967 \text{ (остаток 58)}$$

Таким образом, $27^4 \equiv 58 \pmod{69}$

$$27^5 = 27^4 \times 27 \equiv 58 \times 27 = 2842 \rightarrow 2842 \bmod 69 = 13, \text{ т. к. } 69 \times 41 = 2829 \text{ (остаток 13)}$$

Таким образом, $M_1 = 27^5 \pmod{69} = 13$

Следовательно, $M_1 = 13$, что соответствует букве алфавита «к».

2. $C_2 = 12$

$$12^2 = 144 \rightarrow 144 \bmod 69 = 6, \text{ т. к. } 69 \times 2 = 138 \text{ (остаток 6)}$$

Таким образом, $12^2 \equiv 6 \pmod{69}$

$$12^4 = (12^2)^2 \equiv 6^2 = 36 \rightarrow 36 \bmod 69 = 36$$

Таким образом, $12^4 \equiv 36 \pmod{69}$

$$12^5 = 12^4 \times 12 \equiv 36 \times 12 = 432 \rightarrow 432 \bmod 69 = 18, \text{ т. к. } 69 \times 6 = 414 \text{ (остаток 18)}$$

Таким образом, $M_2 = 12^5 \pmod{69} = 18$

Следовательно, $M_2 = 18$, что соответствует букве алфавита «р».

3. $C_3 = 39$

$$39^2 = 1521 \rightarrow 1521 \bmod 69 = 3, \text{ т. к. } 69 \times 22 = 1518 \text{ (остаток 3)}$$

Таким образом, $39^2 \equiv 3 \pmod{69}$

$$39^4 = (39^2)^2 \equiv 3^2 = 9 \rightarrow 9 \bmod 69 = 9$$

Таким образом, $39^4 \equiv 9 \pmod{69}$



$$39^5 = 39^4 \times 39 \equiv 9 \times 39 = 351 \rightarrow 351 \bmod 69 = 6, \text{ т. к. } 69 \times 5 = 345 \text{ (остаток 6)}$$

$$\text{Таким образом, } M_3 = 39^5 \bmod 69 = 6$$

Следовательно, $M_3 = 6$, что соответствует букве алфавита «е».

4. $C_4 = 44$

$$44^2 = 1936 \rightarrow 1936 \bmod 69 = 4, \text{ т. к. } 69 \times 28 = 1932 \text{ (остаток 4)}$$

$$\text{Таким образом, } 44^2 \equiv 4 \pmod{69}$$

$$44^4 = (44^2)^2 \equiv 4^2 = 16 \rightarrow 16 \bmod 69 = 16$$

$$\text{Таким образом, } 44^4 \equiv 16 \pmod{69}$$

$$44^5 = 44^4 \times 44 \equiv 16 \times 44 = 704 \rightarrow 704 \bmod 69 = 14, \text{ т. к. } 69 \times 10 = 690 \text{ (остаток 14)}$$

$$\text{Таким образом, } M_4 = 44^5 \bmod 69 = 14$$

Следовательно, $M_4 = 14$, что соответствует букве алфавита «м».

5. $C_5 = 39$

Аналогично C_3 , $M_3 = M_5 = 6$, что соответствует букве алфавита «е».

6. $C_6 = 60$

$$60^2 = 3600 \rightarrow 3600 \bmod 69 = 12, \text{ т. к. } 69 \times 52 = 3588 \text{ (остаток 12)}$$

$$\text{Таким образом, } 60^2 \equiv 12 \pmod{69}$$

$$60^4 = (60^2)^2 \equiv 12^2 = 144 \rightarrow 144 \bmod 69 = 6, \text{ т. к. } 69 \times 2 = 138 \text{ (остаток 6)}$$

$$\text{Таким образом, } 60^4 \equiv 6 \pmod{69}$$

$$60^5 = 60^4 \times 60 \equiv 6 \times 60 = 360 \rightarrow 360 \bmod 69 = 15, \text{ т. к. } 69 \times 5 = 345 \text{ (остаток 15)}$$

$$\text{Таким образом, } M_6 = 60^5 \bmod 69 = 15$$

Следовательно, $M_6 = 15$, что соответствует букве алфавита «н».

7. $C_7 = 15$

Аналогично C_2 для первого слова $M_2 = M_7 = 30$, что соответствует букве алфавита «ь».

Соединив расшифрованные буквы, получаем слово «кремень».

В ответе сказано указать два слова расшифрованного сообщения в формате «Первое слово» «Второе слово», соединяя получаем льдистый кремьнь.

Ответ: льдистый кремьнь

**Задача 5 (20 баллов)****Маршруты в Кибертауне**

В футуристическом городе Кибертауне роботы-курьеры доставляют посылки по сложной системе туннелей. Каждый маршрут робота записывается как последовательность команд управления, разделённых точками зарядки *.

Примеры маршрутов:

AB (2 команды без зарядки);

AB*CD (блок из 2 команд, зарядка, блок из 2 команд);

A*B*C (3 блока по 1 команде с зарядками между ними).

Алфавит команд (система управления) поддерживает ровно **K** различных команд (например: A, B, C, D, E, F — движение в разных направлениях).

Маршрут состоит из одного или нескольких блоков команд, блоки разделяются символом зарядки *, каждый блок — это последовательность из 1 или более команд.

Общая длина записи маршрута $\leq N$ символов (включая команды и разделители *), при этом маршрут **не может начинаться с ***, маршрут **не может заканчиваться на ***, два разделителя ** подряд **запрещены** (между зарядками всегда есть хотя бы одна команда).

Найти количество различных корректных маршрутов.

Формат входных данных

Одна строка с двумя целыми числами: **N** и **K** через пробел, **N** — максимальная длина маршрута (в количестве команд, в том числе зарядки), **K** — количество различных команд ($1 \leq N \leq 5000$, $1 \leq K \leq 26$).

Формат выходных данных

Одно целое число — количество корректных маршрутов.

Примеры

стандартный ввод	стандартный вывод
3 2	18

Пояснение к примеру

Например, две команды: A, B.

Корректные маршруты длины ≤ 3 :

без зарядок длины 1: A, B — 2 маршрута;

без зарядок длины 2: AA, AB, BA, BB — 4 маршрута;

без зарядок длины 3: AAA, AAB, ABA, ABB, BAA, BAB, BBA, BBB — 8 маршрутов;

с одной зарядкой (структура 1*1): A*A, A*B, B*A, B*B — 4 маршрута;



с одной зарядкой (структура 2^* или $*2$ невозможны, т.к. нельзя начинать/заканчивать с $*$).

Итого: 18 маршрутов.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных данных, поясните решение. Оцените сложность предложенного алгоритма.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
2 3	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Смотрите решение в первом варианте.



Задача 6 (30 баллов)

Ящики Грю

В коридоре у известного суперзлодея Грю скопилась целая куча ящиков. Все они пронумерованы и расположены вдоль стены в одну линию друг за другом. Грю уже и не помнит, почему он их так пронумеровал, но на крышках этих ящиков есть самые разные номера, в том числе и повторяющиеся.

Будучи в душе перфекционистом, Грю давно хочет навести порядок в коридоре. Под порядком он понимает тот факт, чтобы ящики были расположены так же вдоль стены коридора в одну линию, но уже в порядке – порядке неубывания номеров. И вот, наконец-то сегодня, он решил осуществить свой суперзлодейский план по упорядочиванию ящиков.

Само собой, для этого у него есть целая армия неугомонных миньонов. Грю знает, что, если он даст работу некоторым из них, а остальных проигнорирует, те на него сильно обидятся. По этой причине Грю, вообще-то не отличающийся стандартным мышлением, решил действовать так: он выделит каждому миньону некоторый отрезок из нескольких подряд идущих ящиков и попросит того отсортировать эти ящики по неубыванию. Если миньону достанется такой отрезок из ящиков, что ему придется хотя бы немного их переставить – то миньон будет доволен, однако, если ему достанется уже упорядоченный отрезок, миньон будет считать, что его зря позвали и снова обидится. Чтобы миньоны не подрались, их отрезки не должны пересекаться.

Грю желает выделить отрезки миньонам так, чтобы после их работы весь ряд ящиков так же стал упорядочен по неубыванию – всё прямо как он и задумывал в своих суперзлодейских мечтах.

Теперь пришло время делить фронт работ, то есть набор ящиков на непересекающиеся отрезки. И тут Грю понял, что его супермозг не может решить такую задачу. Следует помочь ему и сказать какое максимально возможное число миньонов он сможет осчастливить, выделив каждому непустой отрезок, внутри которого ящики пока еще не стоят в нужном порядке, а после того, как их поставить в каждом таком отрезке по неубыванию, то и вся последовательность ящиков станет упорядоченной по неубыванию.

Формат входных данных

В первой строке находится одно целое число n – количество ящиков, которые находятся в коридоре у Грю ($1 \leq n \leq 2 \cdot 10^5$).

Во второй строке через пробел указаны целые номера b_i этих ящиков в том порядке, в котором они стоят вдоль стены ($1 \leq b_i \leq 10^{18}$).

Формат выходных данных

Вывести в ответ одно число – **максимальное** количество миньонов,



которых следует привлечь к упорядочиванию ящиков по методу Грю, с условием, что каждому достанется непустой неупорядоченный отрезок из ящиков. После того, как каждый миньон упорядочит свой отрезок, вся последовательность ящиков должна оказаться так же упорядоченной по неубыванию.

Примеры

стандартный ввод	стандартный вывод
5 2 1 4 5 3	2
4 3 3 4 5	0
5 7 1 3 2 2	1
11 3 4 3 2 4 7 8 7 12 12 9	3

Комментарии к примерам

В первом примере следует одному миньону выделить отрезок от первого до второго ящика, а второму миньону от третьего до пятого ящика.

Во втором примере набор ящиков уже упорядочен, и миньонам здесь делать нечего.

В третьем наборе единственный способ – выбрать один отрезок из всех ящиков. Нужен один миньон.

В четвёртом примере нужны три миньона. Первому следует выделить отрезок с первого до шестого (это ящики подписаны 3 4 3 2 4 7), второму с седьмого до восьмого (ящики подписаны 8 7), третьему оставшиеся ящики с девятого по одиннадцатый (на них стоят надписи 12 12 9). После того, как миньоны упорядочат каждый свои ящики, последовательность ящиков станет так же упорядоченной. Существуют и другие способы раздать отрезки в этом примере, однако больше трёх миньонов никогда не понадобится.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных данных. Оцените сложность предложенного алгоритма.



Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
11 4 1 6 3 5 10 7 6 9 11 8	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Смотрите решение в первом варианте.



ВАРИАНТ 3

Задача 1 (5 баллов)

Разведчики

От разведчиков были получены сведения, закодированные в виде электронной таблицы. Определите числа в ячейках C2:C11, которые будут получены в результате копирования в них блока ячеек B1:B10.

В ячейках A1:A5 разместите десятичные результаты вычисления чисел разных систем счисления:

$$A1 = 10_5 - 22_3 + 12_4$$

$$A2 = 22_4 - 22_5 + 10_5$$

$$A3 = 14_5 - 11_5 + 12_3$$

$$A4 = 22_3 - 11_5 + 10_4$$

$$A5 = 13_4 - 12_3 + 11_2$$

В ячейке A6 разместите формулу: = "АВВ".

	A	B	C	D
1		=СУММ(A\$1:\$A6)		
2		=МИН(\$A\$1:\$A6;\$B\$1)		
3		=МАКС(\$A1:\$A6;\$B\$1:B2)		
4		=ЕСЛИ(A\$6=0;A\$4;A\$5)		
5		=СУММЕСЛИ(A1:B\$4;A2)		
6		=СЧЁТЕСЛИ(A1:B\$5;"<A4")		
7		=ЕСЛИ(ИЛИ(B\$3>\$B4;\$B\$5<\$B6);A4;A\$5)		
8		=ЕСЛИ(И(B4<B5;B6>B7);A1;A2)		
9		=ЕСЛИ(B7-\$B\$8>0;B\$7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;B5))		
10		=ЕСЛИ(B8>\$B9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))		
11				
12				

Ответ записать в одну строчку через запятую, начиная с ячейки C2 и заканчивая ячейкой C11.

Решение

Получим значения в ячейках A1:A5 как результат вычисления чисел разных систем счисления:

$$A1 = 10_5 - 22_3 + 12_4 = 5 - 8 + 6 = 3$$

$$A2 = 22_4 - 22_5 + 10_5 = 10 - 12 + 5 = 3$$

$$A3 = 14_5 - 11_5 + 12_3 = 9 - 6 + 5 = 8$$

$$A4 = 22_3 - 11_5 + 10_4 = 8 - 6 + 4 = 6$$

$$A5 = 13_4 - 12_3 + 11_2 = 7 - 5 + 3 = 5$$

Следующим шагом копируем блок ячеек B1:B10 в ячейки C2:C11, в результате получим следующие формулы:



	A	B	C
1	3	=СУММ(A\$1:\$A6)	
2	3	=МИН(\$A\$1:A\$6;B\$1)	=СУММ(B\$1:\$A7)
3	8	=МАКС(\$A1:A\$6;\$B\$1:B2)	=МИН(\$A\$1:B\$6;C\$1)
4	6	=ЕСЛИ(A\$6=0;A\$4;\$A5)	=МАКС(\$A2:B\$6;\$B\$1:C3)
5	5	=СУММЕСЛИ(A1:B\$4;A2)	=ЕСЛИ(B\$6=0;B\$4;\$A6)
6	= "АБВ"	=СЧЁТЕСЛИ(A1:B\$5;"<A4")	=СУММЕСЛИ(B2:C\$4;B3)
7		=ЕСЛИ(ИЛИ(B\$3>\$B4;\$B\$5<\$B6);A4;\$A5)	=СЧЁТЕСЛИ(B2:C\$5;"<A4")
8		=ЕСЛИ(И(B4<\$B5;\$B6>\$B7);A1;A2)	=ЕСЛИ(ИЛИ(C\$3>\$B5;\$B\$5<\$C6);B5;\$B5)
9		=ЕСЛИ(B7-\$B\$8>0;\$B7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;\$B5))	=ЕСЛИ(И(C5<\$C6;\$C7>\$C8);B2;\$B3)
10		=ЕСЛИ(B8>\$B9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))	=ЕСЛИ(C8-\$B\$8>0;\$C7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;\$B5))
11			=ЕСЛИ(C9>\$C9;СЧЁТЕСЛИ(C2:C10;">7");СЧЁТЕСЛИ(C2:C10;"<7"))
12			

Вычисляем значения по формулам в ячейках В1:В10, затем переходим к вычислениям значений в ячейках С2:С11. Получаем следующие результаты:

	A	B	C	D
1	3	3		
2	3	3	3	
3	8	8	0	
4	6	5	12	
5	5	12	5	
6	АБВ	0	8	
7		6	0	
8		3	12	
9		6	8	
10		7	0	
11			5	
12				

Ответ: 3, 0, 12, 5, 8, 0, 12, 8, 0, 5

**Задача 2 (10 баллов)****Ключ от архива**

Историк Петр обнаружил старый архив, защищённый кодовым замком. Документы содержат информацию, зашифрованную четырехразрядным числовым ключом в шестнадцатеричной системе счисления. Петр знает, что каждая цифра числа слева направо была преобразована в двоичную систему счисления и записана четырехразрядным равномерным кодом, т.е. если двоичное представление содержало менее четырёх разрядов, оно дополнялось нулями слева до полного четырёхразрядного кода (например, $716=01112$). Полученные двоичные коды соединили последовательно слева направо.

Таблица истинности этого логического выражения содержит все необходимые значения в результирующем столбце сверху вниз для определения кода:

$$a \vee b \wedge \neg c \oplus d \equiv (a \wedge b \vee d \rightarrow (b \vee \neg c))$$

Строки значений a, b, c, d в таблице истинности идут последовательно, в лексикографическом порядке 0000, 0001, 0010 и т.д. Определите четырёхзначный код архива в шестнадцатеричной системе счисления.

Решение

Составим таблицу истинности

a	b	c	d	$\neg c$	$b \wedge \neg c$	$a \vee b \wedge \neg c$	$a \vee b \wedge \neg c \oplus d$	$a \wedge b$	$a \wedge b \vee d$	$b \vee \neg c$	$a \wedge b \vee d \rightarrow (b \vee \neg c)$	Итого
0	0	0	0	1	0	0	0	0	0	1	1	0
0	0	0	1	1	0	0	1	0	1	1	1	1
0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	1	0	0	0	1	0	1	0	0	0
0	1	0	0	1	1	1	1	0	0	1	1	1
0	1	0	1	1	1	1	0	0	1	1	1	0
0	1	1	0	0	0	0	0	0	0	1	1	0
0	1	1	1	0	0	0	1	0	1	1	1	1
1	0	0	0	1	0	1	1	0	0	1	1	1
1	0	0	1	1	0	1	0	0	1	1	1	0
1	0	1	0	0	0	1	1	0	0	0	0	1
1	0	1	1	0	0	1	0	0	1	0	0	1
1	1	0	0	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	1	1	1	1	0
1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	1	1	0	0	1	0	1	1	1	1	0

Двоичный код в последнем столбце можно разбить на четыре кодовых слова длины 4 числа следующим образом: 0100, 1001, 1011, 1010.

Таким образом получаем следующий шестнадцатеричный код: $49BA_{16}$ или $49(11)(10)_{16}$

Ответ: $49BA_{16}$ или $49(11)(10)_{16}$



Задача 3 (15 баллов)

Циклический возврат камеры наблюдения

Исполнитель Камера используется для управления камерой наблюдения, которая может вращаться ограниченно по вертикали и без ограничений по горизонтали. Исполнитель Камера может выполнять следующие команды:

1) Повернуть на (a, b) , где a, b – целые числа. Эта команда поворачивает камеру на величину вверх по вертикали на угол a , заданный в градусах, и поворачивает его против часовой стрелки на угол b , заданный в градусах. Например, команда Повернуть на $(-2, -30)$ опустит камеру на 2 градуса по вертикали вниз и повернет его на угол 30 по горизонтали по часовой стрелке.

2) Цикл:

ПОВТОРИ n

последовательность команд

КОНЕЦ ПОВТОРИ

Этот цикл означает, что последовательность команд будет выполняться n раз.

Исполнителю Камера был дан для исполнения следующий алгоритм (буквами a, b и n обозначены неизвестные целые числа):

НАЧАЛО

повернуть на $(60, 180)$

ПОВТОРИ n

повернуть на (a, b)

повернуть на $(2, -60)$

КОНЕЦ ПОВТОРИ

повернуть на $(40, -60)$

КОНЕЦ

Чему равно максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов. Чему равно минимально возможное число полных оборотов (360 градусов), которое при этом совершит камера?

Чему равны значения n , если после выполнения алгоритма камера вернулась в то же состояние, что было до начала выполнения алгоритма, совершив 5 полных оборотов? Чему, при этом равны параметры a и b ?

Решение

Значение параметров исполнителя Камера в результате выполнения алгоритма рассчитывается по формуле:

$$l - l_0 = 60 + n(a + 2) + 40$$



$$f - f_0 = 180 + n(b - 60) - 60$$

В этих уравнениях n – количество повторений команд ПОВТОРИ, l и l_0 – конечное и начальное значение угла вертикального поворота камеры, f и f_0 – конечное и начальное значение угла горизонтального поворота камеры.

Так как камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, $l - l_0 = 0$, а $f - f_0 = 360k$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Знак $+$ соответствует оборотам против часовой стрелки, а знак $-$ соответствует оборотам по часовой стрелке.

С учетом этих условий, получим систему уравнений:

$$60 + n(a + 2) + 40 = 0$$

$$180 + n(b - 60) - 60 = 360k$$

После преобразований, получим:

$$n(a + 2) = -100$$

$$n(b - 60) = 360k - 120$$

Нужно найти натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число n должно быть одновременно делителем чисел 100 и $360k - 120$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Наибольшим общим делителем для этих выражений будет число 100. Оно будет соответствовать максимально возможному значению n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, при условии, что $360k - 120$ будет кратно 100. Минимальное число оборотов равно 2, что соответствует $k = 2$.

При совершении 5 полных оборотов возможны два варианта, в зависимости от направления вращения:

$$n(b - 60) = 360 \cdot 5 - 120 = 1680$$

$$n(b - 60) = 360 \cdot (-5) - 120 = -1920$$

Нужно найти наибольшее натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b – наибольшее число, которое является делителем чисел 100 и 1920 или 100 и 1680. В обоих случаях наибольший общий делитель $n = 20$. Рассчитав значения a и b , получим:

$$a = -100/20 - 2 = -7$$

$$b+ = 1680/20 + 60 = 144 \text{ – при оборотах против часовой стрелки}$$

$$b- = -1920/20 + 60 = -36 \text{ – при оборотах по часовой стрелке}$$

Остальные значения n представляют собой делители числа 20: 1, 2, 4, 5, 10. Для них расчет значений a и b производится по аналогичным формулам.



Ответ: максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 100, при этом минимально возможное число полных оборотов равно 2. При совершении 5 полных оборотов максимальное $n = 20$, $a = -7$, $b = -36$ при оборотах по часовой стрелке и $b = 144$ при оборотах против часовой стрелки.



Задача 4 (20 баллов)

Геологический шифр

В ходе геологоразведочных работ в Арктической зоне РФ была обнаружена неизвестная науке горная порода с уникальными кристаллическими свойствами. Чтобы сохранить приоритет открытия в тайне от конкурентов, старший геолог Пётр передал для анализа коллеге Марии зашифрованные данные, содержащие предварительное название породы из двух слов.

Мария получила два файла:

- 1) rock_sample.dat – файл с основным зашифрованным отчётом.
- 2) thin_section.png – микрофотография тонкого среза породы, в которую Петром был внедрён секретный ключ методом стеганографии, имитирующим природные включения.

Алгоритм действий для Марии выглядит следующим образом:

1) Извлечь скрытые данные из изображения thin_section.png. Для этого необходимо проанализировать младшие биты цветовых каналов пикселей, что позволит получить битовую последовательность.

2) Эту битовую последовательность необходимо декодировать. В результате она должна превратиться в строку в формате JSON без кавычек, которая и содержит секретные ключи для алгоритма RSA. В данной задаче строка в формате JSON без кавычек выглядит следующим образом: {ключ1:значение1,ключ2: значение2,ключ3:значение3}.

3) Используя полученные ключи, можно приступить к дешифровке основного отчёта, хранящегося в файле rock_sample.dat.

В результате анализа младших битов цветовых каналов пикселей изображения thin_section.png Мария получила следующую строку:

110000001110011101011111011010000011001111100110011110111101000
101000110011111010110001

Пётр, известный своей любовью к горным треккигам, для защиты информации в файле thin_section.png применяет специальный «Горный код» стеганографии. Его принцип основан на использовании неравномерного кодирования, где частым символам соответствуют короткие коды, что минимизирует общий размер секретного сообщения. Дополнительный уровень защиты обеспечивают «мусорные» символы – специальные ловушки, встроенные в код. При расшифровке легитимный получатель должен распознать и отфильтровать этот мусор, чтобы получить исходные данные. Таблица используемых Петром кодов представлена ниже.

Символ	Код	Символ	Код
{	110000	7	111100
}	110001	8	111101
"	110010	9	111110



Символ	Код	Символ	Код
:	110011	n	000
,	110100	e	001
0	110101	d	010
1	110110	p	0110
2	110111	q	0111
3	111000	x	1000
4	111001	z	1001
5	111010	y	1010
6	111011		

Неравномерный код – это код, в котором кодовые слова, например, последовательности битов, имеют разную длину.

Например, при декодировании битовой последовательности 000001010 с помощью вышепредставленной таблицы, процесс декодирования будет заключаться в следующем:

1) Читаем биты слева-направо: 0 → 00 → 000.

000 есть в таблице? Да, это n.

Результат: n.

2) Продолжаем со следующего бита: 0 → 00 → 001.

001 есть в таблице? Да, это e.

Результат: e.

3) Продолжаем со следующего бита: 0 → 01 → 010.

010 есть в таблице? Да, это d.

Результат: d.

Расшифрованное сообщение n e d.

Файл rock_sample.dat представляет собой последовательность десятичных чисел, каждая из которых – зашифрованная буква передаваемого сообщения. Исходный текст отчёта перед шифрованием был преобразован в числа по правилу: каждый символ кодировался своим порядковым номером в алфавите (a=1, б=2, ..., я=33). Для шифрования Петром использовалась формула: $C_i = M_i^e \bmod n$,

где C_i – i-й элемент последовательности шифротекста C;

M_i – i-й элемент последовательности шифра M;

e – открытая экспонента – параметр открытого ключа RSA – число, используемое при шифровании в качестве степени;

n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.

Для дешифрования должна использоваться формула $M_i = C_i^d \bmod n$,

где M_i – i-й элемент последовательности шифра M;

C_i – i-й элемент последовательности шифротекста C;

d – секретная экспонента – параметр закрытого ключа RSA – число, используемое при дешифровке в качестве степени;

n – модуль операции взятия остатка;



mod – операция взятия остатка от деления.

Совокупность значений (e, n) и (d, n) является открытым и закрытым ключами, используемыми при шифровании и дешифровании соответственно.

Например, RSA-шифрование с открытым ключом: $(e, n) = (17, 3233)$ – и закрытым ключом: $(d, n) = (2753, 3233)$ заключается в следующем.

При шифровании сообщения $M = 65$ отправляемый шифротекст $C = M^e \text{mod } n = 65^{17} \text{mod } 3233 = 2790$

При дешифровке полученного шифротекста $C = 2790$ восстановленное сообщение $M = C^d \text{mod } n = 2790^{2753} \text{mod } 3233 = 65$.

Примечание

Чтобы вычислить выражение вида $M = C^d \text{mod } n$, иногда невыгодно сначала находить большое число C^d , ведь оно может быть огромным. Удобнее использовать алгоритм быстрого возведения в степень по модулю.

Идея алгоритма:

1. Разложить показатель степени на степени двойки.
2. Последовательно возводить число в квадрат.
3. После каждого шага брать остаток по модулю.
4. Умножать только те степени, которые нужны по разложению.

Например, если нужно найти значение выражения $11^5 \text{mod } 9$, то степень 5 можно представить как сумму степеней 4 и 1: $5 = 4 + 1$.

Следовательно:

1. Находим остаток: $11^2 \text{mod } 9 = 4$, т.к. $9 \times 13 = 117$ (остаток 4).

2. Используем найденный остаток, чтобы получить $11^4 \text{mod } 9$.

$11^4 \text{mod } 9 = (11^2)^2 \text{mod } 9 \equiv 4^2 \text{mod } 9 = 7$, т.к. $9 \times 1 = 9$ (остаток 7)

3. Перемножаем найденный остаток с 11 – и снова берём остаток.

$11^5 \text{mod } 9 = (11^4 \times 11) \text{mod } 9 \equiv (7 \times 11) \text{mod } 9 = 77 \text{mod } 9 = 5$, т.к. $9 \times 8 = 72$ (остаток 5)

Таким образом, $11^5 \text{mod } 9 = 5$.

Используя данные, извлечённые из стеганографического сообщения, расшифруйте `rock_sample.dat` и узнайте предварительное название новой горной породы. Содержимое `rock_sample.dat`:

Первое слово – 25 37 8 19 37 39 19; второе слово – 9 22 22 17 25 20 41.

В ответе укажите два слова расшифрованного сообщения в формате «Первое слово» «Второе слово» (например, Медное золото).

Решение

На первом этапе необходимо определить секретный ключ, который был внедрен Петром методом стеганографии в файл `thin_section.png`.

В задаче сказано, что Пётр использовал для шифрования специальный вод, который основан на использовании неравномерного кодирования. Алгоритм дешифровки в данном случае заключается в следующем: читаем



строку по битам слева направо, пошагово накапливая последовательность, пока она не совпадёт с одним из кодов в таблице. Как только совпадение найдено, записываем символ и начинаем формировать следующую последовательность заново.

В условии сказано, что Мария в результате анализа данного изображения получила битовую строку:

110000001110011101011110110100000110011111001100111101111010
00101000110011111010110001

Начнём процесс декодирования. Если код не найден сразу (например, при чтении первой 1), то показан процесс накопления последовательности: $1 \rightarrow 11 \rightarrow 110$ и т.д., до момента совпадения с записью в таблице кодов, используемых Петром при шифровании. Максимальная длина кодового слова в таблице кодировки – 6 бит (например, 110000, 110011, 111110 и т.д.), поэтому в процессе накопления битовой последовательности последовательности длиннее 6 символов не рассматриваются.

Шаг	Индекс начала	Биты (извлечённые)	Найденный код	Символ
1	0	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110000$	110000	{
2	6	$0 \rightarrow 00 \rightarrow 001$	001	e
3	9	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
4	15	$1 \rightarrow 10 \rightarrow 101 \rightarrow 1010$	1010	y
5	19	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1111 \rightarrow 11111 \rightarrow 111110$	111110	9
6	25	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1101 \rightarrow 11010 \rightarrow 110100$	110100	,
7	31	$0 \rightarrow 00 \rightarrow 000$	000	n
8	34	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
9	40	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1110 \rightarrow 11100 \rightarrow 111001$	111001	4
10	46	$1 \rightarrow 10 \rightarrow 100 \rightarrow 1001$	1001	z
11	50	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111011$	111011	6
12	56	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1101 \rightarrow 11010 \rightarrow 110100$	110100	,
13	62	$0 \rightarrow 01 \rightarrow 010$	010	d
14	65	$1 \rightarrow 10 \rightarrow 100 \rightarrow 1000$	1000	x
15	69	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
16	75	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111010$	111010	5
17	81	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110001$	110001	}

Собирая полученные символы в порядке их дешифровки, получаем точную расшифровку битовой строки: {e:y9,n:4z6,dx:5}. В условии задачи говорится про «мусорные символы» – ловушки, которые Пётр вставляет в сообщение. В таблице кодов есть набор букв-«включений» x, y, z (и т.д.), и



они выступают как мусор в конкретном сообщении: они появляются внутри ключей/значений, но не несут смысла для конечного JSON.

Если удалить все мусорные буквы x, y, z из полученной строки {e:y9,n:4z6,dx:5}, то получим корректную JSON-подобную структуру без кавычек: {e:9,n:46,d:5}. Следовательно, получены данные для расшифровки текстового сообщения из файла rock_sample.dat, а именно модуль $n=46$ и секретная экспонента $d=5$.

Файл rock_sample.dat содержит две последовательности целых чисел, где каждое число – зашифрованный символ шифротекста:

Первое слово: 25 37 8 19 37 39 19

Второе слово: 9 22 22 17 25 20 41

Из условия задачи воспользуемся формулой дешифрования: $M_i = C_i^d \bmod n$,

где M_i – i -й элемент последовательности шифра M (восстановленный номер буквы (1 = а, 2 = б, ..., 33 = я);

C_i – i -й элемент последовательности шифротекста C ;

d – секретная экспонента;

n – модуль операции взятия остатка;

$\bmod$ – операция взятия остатка от деления.

Дешифруем первое слово: $C = [25 \ 37 \ 8 \ 19 \ 37 \ 39 \ 19]$. Можно производить дешифровку напрямую возводя числа в степень $d=5$ или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 25$

$$25^2 = 625 \rightarrow 625 \bmod 46 = 27, \text{ т. к. } 46 \times 13 = 598 \text{ (остаток 27)}$$

$$\text{Таким образом, } 25^2 \equiv 27 \pmod{46}$$

$$25^4 = (25^2)^2 \equiv 27^2 = 729 \rightarrow 729 \bmod 46 = 39, \text{ т. к. } 46 \times 15 = 690 \text{ (остаток 39)}$$

$$\text{Таким образом, } 25^4 \equiv 39 \pmod{46}$$

$$25^5 = 25^4 \times 25 \equiv 39 \times 25 = 975 \rightarrow 975 \bmod 46 = 9, \text{ т. к. } 46 \times 21 = 966 \text{ (остаток 9)}$$

$$\text{Таким образом, } M_1 = 25^5 \pmod{46} = 9$$

Следовательно, $M_1 = 9$, что соответствует букве алфавита «з».

2. $C_2 = 37$

$$37^2 = 1369 \rightarrow 1369 \bmod 46 = 35, \text{ т. к. } 46 \times 29 = 1334 \text{ (остаток 35)}$$

$$\text{Таким образом, } 37^2 \equiv 35 \pmod{46}$$

$$37^4 = (37^2)^2 \equiv 35^2 = 1225 \rightarrow 1225 \bmod 46 = 29, \text{ т. к. } 46 \times 26 = 1196 \text{ (остаток 29)}$$

$$\text{Таким образом, } 37^4 \equiv 29 \pmod{46}$$

$$37^5 = 37^4 \times 37 \equiv 29 \times 37 = 1073 \rightarrow 1073 \bmod 46 = 15, \text{ т. к. } 46 \times 23 = 1058 \text{ (остаток 15)}$$

$$\text{Таким образом, } M_2 = 37^5 \pmod{46} = 15$$

Следовательно, $M_2 = 15$, что соответствует букве алфавита «н».

3. $C_3 = 8$



$$8^2 = 64 \rightarrow 64 \bmod 46 = 18, \text{ т. к. } 46 \times 1 = 46 \text{ (остаток 18)}$$

Таким образом, $8^2 \equiv 18 \pmod{46}$

$$8^4 = (8^2)^2 \equiv 18^2 = 324 \rightarrow 324 \bmod 46 = 2, \text{ т. к. } 46 \times 7 = 322 \text{ (остаток 2)}$$

Таким образом, $8^4 \equiv 2 \pmod{46}$

$$8^5 = 8^4 \times 8 \equiv 2 \times 8 = 16 \rightarrow 16 \bmod 46 = 16$$

Таким образом, $M_3 = 8^5 \pmod{46} = 16$

Следовательно, $M_3 = 16$, что соответствует букве алфавита «о».

4. $C_4 = 19$

$$19^2 = 361 \rightarrow 361 \bmod 46 = 39, \text{ т. к. } 46 \times 7 = 322 \text{ (остаток 39)}$$

Таким образом, $19^2 \equiv 39 \pmod{46}$

$$19^4 = (19^2)^2 \equiv 39^2 = 1521 \rightarrow 1521 \bmod 46 = 3, \text{ т. к. } 46 \times 33 = 1518 \text{ (остаток 3)}$$

Таким образом, $19^4 \equiv 3 \pmod{46}$

$$19^5 = 19^4 \times 19 \equiv 3 \times 19 = 57 \rightarrow 57 \bmod 46 = 11, \text{ т. к. } 46 \times 1 = 46 \text{ (остаток 11)}$$

Таким образом, $M_4 = 19^5 \pmod{46} = 11$

Следовательно, $M_4 = 11$, что соответствует букве алфавита «й».

5. $C_5 = 37$

Аналогично $C_2, M_2 = M_5 = 15$, что соответствует букве алфавита «н».

6. $C_6 = 39$

$$39^2 = 1521 \rightarrow 1521 \bmod 46 = 3, \text{ т. к. } 46 \times 33 = 1518 \text{ (остаток 3)}$$

Таким образом, $39^2 \equiv 3 \pmod{46}$

$$39^4 = (39^2)^2 \equiv 3^2 = 9 \rightarrow 9 \bmod 46 = 9$$

Таким образом, $39^4 \equiv 9 \pmod{46}$

$$39^5 = 39^4 \times 39 \equiv 9 \times 39 = 351 \rightarrow 351 \bmod 46 = 29, \text{ т. к. } 46 \times 7 = 322 \text{ (остаток 29)}$$

Таким образом, $M_6 = 39^5 \pmod{46} = 29$

Следовательно, $M_6 = 29$, что соответствует букве алфавита «ы».

7. $C_7 = 19$

Аналогично $C_4, M_4 = M_7 = 11$, что соответствует букве алфавита «й».

Соединив расшифрованные буквы, получаем слово «знойный».

Дешифруем второе слово: $C = [9 \ 22 \ 22 \ 17 \ 25 \ 20 \ 41]$. Можно производить дешифровку напрямую возводя числа в степень $d=5$ или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 9$

$$9^2 = 81 \rightarrow 81 \bmod 46 = 35, \text{ т. к. } 46 \times 1 = 46 \text{ (остаток 35)}$$

Таким образом, $9^2 \equiv 35 \pmod{46}$

$$9^4 = (9^2)^2 \equiv 35^2 = 1225 \rightarrow 1225 \bmod 46 = 29, \text{ т. к. } 46 \times 26 = 1196 \text{ (остаток 29)}$$

Таким образом, $9^4 \equiv 29 \pmod{46}$



$$9^5 = 9^4 \times 9 \equiv 29 \times 9 = 261 \rightarrow 261 \bmod 46 = 31, \text{ т. к. } 46 \times 5 = 230 \text{ (остаток 31)}$$

$$\text{Таким образом, } M_1 = 9^5 \bmod 46 = 31$$

Следовательно, $M_1 = 31$, что соответствует букве алфавита «э».

2. $C_2 = 22$

$$22^2 = 484 \rightarrow 484 \bmod 46 = 24, \text{ т. к. } 46 \times 10 = 460 \text{ (остаток 24)}$$

$$\text{Таким образом, } 22^2 \equiv 24 \pmod{46}$$

$$22^4 = (22^2)^2 \equiv 24^2 = 576 \rightarrow 576 \bmod 46 = 24, \text{ т. к. } 46 \times 12 = 552 \text{ (остаток 24)}$$

$$\text{Таким образом, } 22^4 \equiv 24 \pmod{46}$$

$$22^5 = 22^4 \times 22 \equiv 24 \times 22 = 528 \rightarrow 528 \bmod 46 = 22, \text{ т. к. } 46 \times 11 = 506 \text{ (остаток 22)}$$

$$\text{Таким образом, } M_2 = 22^5 \bmod 46 = 22$$

Следовательно, $M_2 = 22$, что соответствует букве алфавита «ф».

3. $C_3 = 22$

Аналогично C_2 , $M_2 = M_3 = 22$, что соответствует букве алфавита «ф».

4. $C_4 = 17$

$$17^2 = 289 \rightarrow 289 \bmod 46 = 13, \text{ т. к. } 46 \times 6 = 276 \text{ (остаток 13)}$$

$$\text{Таким образом, } 17^2 \equiv 13 \pmod{46}$$

$$17^4 = (17^2)^2 \equiv 13^2 = 169 \rightarrow 169 \bmod 46 = 31, \text{ т. к. } 46 \times 3 = 138 \text{ (остаток 31)}$$

$$\text{Таким образом, } 17^4 \equiv 31 \pmod{46}$$

$$17^5 = 17^4 \times 17 \equiv 31 \times 17 = 527 \rightarrow 527 \bmod 46 = 21, \text{ т. к. } 46 \times 11 = 506 \text{ (остаток 21)}$$

$$\text{Таким образом, } M_4 = 17^5 \bmod 46 = 21$$

Следовательно, $M_4 = 21$, что соответствует букве алфавита «у».

5. $C_5 = 25$

Аналогично C_1 для первого слова, $M_1 = M_5 = 9$, что соответствует букве алфавита «з».

6. $C_6 = 20$

$$20^2 = 400 \rightarrow 400 \bmod 46 = 32, \text{ т. к. } 46 \times 8 = 368 \text{ (остаток 32)}$$

$$\text{Таким образом, } 20^2 \equiv 32 \pmod{46}$$

$$20^4 = (20^2)^2 \equiv 32^2 = 1024 \rightarrow 1024 \bmod 46 = 12, \text{ т. к. } 46 \times 22 = 1012 \text{ (остаток 12)}$$

$$\text{Таким образом, } 20^4 \equiv 12 \pmod{46}$$

$$20^5 = 20^4 \times 20 \equiv 12 \times 20 = 240 \rightarrow 240 \bmod 46 = 10, \text{ т. к. } 46 \times 5 = 230 \text{ (остаток 10)}$$

$$\text{Таким образом, } M_6 = 20^5 \bmod 46 = 10$$

Следовательно, $M_6 = 10$, что соответствует букве алфавита «и».

7. $C_7 = 41$

$$41^2 = 1681 \rightarrow 1681 \bmod 46 = 25, \text{ т. к. } 46 \times 36 = 1656 \text{ (остаток 25)}$$

$$\text{Таким образом, } 41^2 \equiv 25 \pmod{46}$$

$$41^4 = (41^2)^2 \equiv 25^2 = 625 \rightarrow 625 \bmod 46 = 27, \text{ т. к. } 46 \times 13 = 598 \text{ (остаток 27)}$$

$$\text{Таким образом, } 41^4 \equiv 27 \pmod{46}$$



$$41^5 = 41^4 \times 41 \equiv 27 \times 41 = 1107 \rightarrow 1107 \bmod 46 = 3, \text{ т.к. } 46 \times 24 = 1104 \text{ (остаток 3)}$$

$$\text{Таким образом, } M_7 = 41^5 \bmod 46 = 3$$

Следовательно, $M_7 = 3$, что соответствует букве алфавита «в».

Соединив расшифрованные буквы, получаем слово «эффузив».

В ответе сказано указать два слова расшифрованного сообщения в формате «Первое слово» «Второе слово», соединяя получаем знойный эффузив.

Ответ: знойный эффузив

**Задача 5 (20 баллов)****Маршруты в Кибертауне**

В футуристическом городе Кибертауне роботы-курьеры доставляют посылки по сложной системе туннелей. Каждый маршрут робота записывается как последовательность команд управления, разделённых точками зарядки *.

Примеры маршрутов:

AB (2 команды без зарядки);

AB*CD (блок из 2 команд, зарядка, блок из 2 команд);

A*B*C (3 блока по 1 команде с зарядками между ними).

Алфавит команд (система управления) поддерживает ровно **K** различных команд (например: A, B, C, D, E, F — движение в разных направлениях).

Маршрут состоит из одного или нескольких блоков команд, блоки разделяются символом зарядки *, каждый блок — это последовательность из 1 или более команд.

Общая длина записи маршрута $\leq N$ символов (включая команды и разделители *), при этом маршрут **не может начинаться с ***, маршрут **не может заканчиваться на ***, два разделителя ** подряд **запрещены** (между зарядками всегда есть хотя бы одна команда).

Найти количество различных корректных маршрутов.

Формат входных данных

Одна строка с двумя целыми числами: **N** и **K** через пробел, **N** — максимальная длина маршрута (в количестве команд, в том числе зарядки), **K** — количество различных команд ($1 \leq N \leq 5000$, $1 \leq K \leq 26$).

Формат выходных данных

Одно целое число — количество корректных маршрутов.

Примеры

стандартный ввод	стандартный вывод
3 2	18

Пояснение к примеру

Например, две команды: A, B.

Корректные маршруты длины ≤ 3 :

без зарядок длины 1: A, B — 2 маршрута;

без зарядок длины 2: AA, AB, BA, BB — 4 маршрута;

без зарядок длины 3: AAA, AAB, ABA, ABB, BAA, BAB, BBA, BBB — 8 маршрутов;



с одной зарядкой (структура $1*1$): $A*A$, $A*B$, $B*A$, $B*B$ — 4 маршрута;
с одной зарядкой (структура $2*$ или $*2$ невозможны, т.к. нельзя начинать/заканчивать с $*$).

Итого: 18 маршрутов.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных данных, поясните решение. Оцените сложность предложенного алгоритма.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
2 4	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Смотрите решение в первом варианте.



Задача 6 (30 баллов)

Ящики Грю

В коридоре у известного суперзлодея Грю скопилась целая куча ящиков. Все они пронумерованы и расположены вдоль стены в одну линию друг за другом. Грю уже и не помнит, почему он их так пронумеровал, но на крышках этих ящиков есть самые разные номера, в том числе и повторяющиеся.

Будучи в душе перфекционистом, Грю давно хочет навести порядок в коридоре. Под порядком он понимает тот факт, чтобы ящики были расположены так же вдоль стены коридора в одну линию, но уже в порядке – порядке неубывания номеров. И вот, наконец-то сегодня, он решил осуществить свой суперзлодейский план по упорядочиванию ящиков.

Само собой, для этого у него есть целая армия неугомонных миньонов. Грю знает, что, если он даст работу некоторым из них, а остальных проигнорирует, те на него сильно обидятся. По этой причине Грю, вообще-то не отличающийся стандартным мышлением, решил действовать так: он выделит каждому миньону некоторый отрезок из нескольких подряд идущих ящиков и попросит того отсортировать эти ящики по неубыванию. Если миньону достанется такой отрезок из ящиков, что ему придется хотя бы немного их переставить – то миньон будет доволен, однако, если ему достанется уже упорядоченный отрезок, миньон будет считать, что его зря позвали и снова обидится. Чтобы миньоны не подрались, их отрезки не должны пересекаться.

Грю желает выделить отрезки миньонам так, чтобы после их работы весь ряд ящиков так же стал упорядочен по неубыванию – всё прямо как он и задумывал в своих суперзлодейских мечтах.

Теперь пришло время делить фронт работ, то есть набор ящиков на непересекающиеся отрезки. И тут Грю понял, что его супермозг не может решить такую задачу. Следует помочь ему и сказать какое максимально возможное число миньонов он сможет осчастливить, выделив каждому непустой отрезок, внутри которого ящики пока еще не стоят в нужном порядке, а после того, как их поставить в каждом таком отрезке по неубыванию, то и вся последовательность ящиков станет упорядоченной по неубыванию.

Формат входных данных

В первой строке находится одно целое число n – количество ящиков, которые находятся в коридоре у Грю ($1 \leq n \leq 2 \cdot 10^5$).

Во второй строке через пробел указаны целые номера b_i этих ящиков в том порядке, в котором они стоят вдоль стены ($1 \leq b_i \leq 10^{18}$).

Формат выходных данных

Вывести в ответ одно число – **максимальное** количество миньонов,



которых следует привлечь к упорядочиванию ящиков по методу Грю, с условием, что каждому достанется непустой неупорядоченный отрезок из ящиков. После того, как каждый миньон упорядочит свой отрезок, вся последовательность ящиков должна оказаться так же упорядоченной по неубыванию.

Примеры

стандартный ввод	стандартный вывод
5 2 1 4 5 3	2
4 3 3 4 5	0
5 7 1 3 2 2	1
11 3 4 3 2 4 7 8 7 12 12 9	3

Комментарии к примерам

В первом примере следует одному миньону выделить отрезок от первого до второго ящика, а второму миньону от третьего до пятого ящика.

Во втором примере набор ящиков уже упорядочен, и миньонам здесь делать нечего.

В третьем наборе единственный способ – выбрать один отрезок из всех ящиков. Нужен один миньон.

В четвёртом примере нужны три миньона. Первому следует выделить отрезок с первого до шестого (это ящики подписаны 3 4 3 2 4 7), второму с седьмого до восьмого (ящики подписаны 8 7), третьему оставшиеся ящики с девятого по одиннадцатый (на них стоят надписи 12 12 9). После того, как миньоны упорядочат каждый свои ящики, последовательность ящиков станет так же упорядоченной. Существуют и другие способы раздать отрезки в этом примере, однако больше трёх миньонов никогда не понадобится.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных данных. Оцените сложность предложенного алгоритма.



Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
11 4 6 1 3 5 10 7 6 9 11 8	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Смотрите решение в первом варианте.



ВАРИАНТ 4

Задача 1 (5 баллов)

Разведчики

От разведчиков были получены сведения, закодированные в виде электронной таблицы. Определите числа в ячейках C2:C11, которые будут получены в результате копирования в них блока ячеек B1:B10.

В ячейках A1:A5 разместите десятичные результаты вычисления чисел разных систем счисления:

$$A1 = 14_5 - 12_5 + 12_3$$

$$A2 = 22_3 - 14_5 + 10_4$$

$$A3 = 13_4 - 12_3 + 11_2$$

$$A4 = 10_5 - 22_3 + 12_4$$

$$A5 = 22_4 - 22_5 + 10_5$$

В ячейке A6 разместите формулу: = "АБВ".

	A	B	C	D
1		=СУММ(A\$1;\$A6)		
2		=МИН(\$A\$1:A\$6;B\$1)		
3		=МАКС(\$A1:A\$6;\$B\$1:B2)		
4		=ЕСЛИ(A\$6=0;A\$4;A\$5)		
5		=СУММЕСЛИ(A1:B\$4;A2)		
6		=СЧЁТЕСЛИ(A1:B\$5;"<A4")		
7		=ЕСЛИ(ИЛИ(B\$3>\$B4;\$B\$5<\$B6);A4;A\$5)		
8		=ЕСЛИ(И(B4<B5;B6>B7);A1;A2)		
9		=ЕСЛИ(B7-\$B\$8>0;B\$7;ЕСЛИ(\$B\$6-\$B5;\$B\$6;B5))		
10		=ЕСЛИ(B8>B\$9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))		
11				
12				

Ответ записать в одну строчку через запятую, начиная с ячейки C2 и заканчивая ячейкой C11.

Решение

Получим значения в ячейках A1:A5 как результат вычисления чисел разных систем счисления:

$$A1 = 14_5 - 12_5 + 12_3 = 9 - 7 + 5 = 7$$

$$A2 = 22_3 - 14_5 + 10_4 = 8 - 9 + 4 = 3$$

$$A3 = 13_4 - 12_3 + 11_2 = 7 - 5 + 3 = 5$$

$$A4 = 10_5 - 22_3 + 12_4 = 5 - 8 + 6 = 3$$

$$A5 = 22_4 - 22_5 + 10_5 = 10 - 12 + 5 = 3$$

Следующим шагом копируем блок ячеек B1:B10 в ячейки C2:C11, в результате получим следующие формулы:



	A	B	C
1	7	=СУММ(A\$1;\$A6)	
2	3	=МИН(\$A\$1:A\$6;B\$1)	=СУММ(B\$1;\$A7)
3	5	=МАКС(\$A1:A\$6;B\$1:B2)	=МИН(\$A\$1:B\$6;C\$1)
4	3	=ЕСЛИ(A\$6=0;A\$4;A\$5)	=МАКС(\$A2:B\$6;B\$1:C3)
5	3	=СУММЕСЛИ(A1:B\$4;A2)	=ЕСЛИ(B\$6=0;B\$4;A6)
6	= "АБВ"	=СЧЁТЕСЛИ(A1:B\$5;"<A4")	=СУММЕСЛИ(B2:C\$4;B3)
7		=ЕСЛИ(ИЛИ(B\$3>B\$4;B\$5<B\$6);A4;A\$5)	=СЧЁТЕСЛИ(B2:C\$5;"<A4")
8		=ЕСЛИ(И(B4<B5;B6>B7);A1;A2)	=ЕСЛИ(ИЛИ(C\$3>B\$5;B\$5<C\$6);B5;B\$5)
9		=ЕСЛИ(B7-\$B\$8>0;B\$7;ЕСЛИ(\$B\$6-\$B\$5;B\$6;B\$5))	=ЕСЛИ(И(C5<C6;C7>C8);B2;B3)
10		=ЕСЛИ(B8>B\$9;СЧЁТЕСЛИ(B1:B9;">7");СЧЁТЕСЛИ(B1:B9;"<7"))	=ЕСЛИ(C8-\$B\$8>0;C\$7;ЕСЛИ(\$B\$6-\$B\$5;B\$6;C6))
11			=ЕСЛИ(C9>C\$9;СЧЁТЕСЛИ(C2:C10;">7");СЧЁТЕСЛИ(C2:C10;"<7"))
12			

Вычисляем значения по формулам в ячейках В1:В10, затем переходим к вычислениям значений в ячейках С2:С11. Получаем следующие результаты:

	A	B	C	D
1	7	7		
2	3	3	7	
3	5	7	0	
4	3	3	12	
5	3	12	3	
6	АБВ	0	14	
7		3	0	
8		3	12	
9		0	7	
10		1	0	
11			4	
12				

Ответ: 7, 0, 12, 3, 14, 0, 12, 7, 0, 4

**Задача 2 (10 баллов)****Ключ от архива**

Историк Петр обнаружил старый архив, защищённый кодовым замком. Документы содержат информацию, зашифрованную четырехразрядным числовым ключом в шестнадцатеричной системе счисления. Петр знает, что каждая цифра числа слева направо была преобразована в двоичную систему счисления и записана четырехразрядным равномерным кодом, т.е. если двоичное представление содержало менее четырёх разрядов, оно дополнялось нулями слева до полного четырёхразрядного кода (например, $716=01112$). Полученные двоичные коды соединили последовательно слева направо.

Таблица истинности этого логического выражения содержит все необходимые значения в результирующем столбце сверху вниз для определения кода:

$$a \wedge b \vee \neg c \rightarrow (b \wedge d \vee a \oplus c \wedge a) \equiv d$$

Строки значений a, b, c, d в таблице истинности идут последовательно, в лексикографическом порядке 0000, 0001, 0010 и т.д. Определите четырёхзначный код архива в шестнадцатеричной системе счисления.

Решение

Составим таблицу истинности

a	b	c	d	$a \wedge b$	$\neg c$	$a \wedge b \vee \neg c$	$b \wedge d$	$b \wedge d \vee a$	$c \wedge a$	$b \wedge d \vee a \oplus c \wedge a$	$a \wedge b \vee \neg c \rightarrow (b \wedge d \vee a \oplus c \wedge a)$	Итого
0	0	0	0	0	1	1	0	0	0	0	0	1
0	0	0	1	0	1	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	1	0	0	0	0	0	0	0	1	1
0	1	0	0	0	1	1	0	0	0	0	0	1
0	1	0	1	0	1	1	1	1	0	1	1	1
0	1	1	0	0	0	0	0	0	0	0	1	0
0	1	1	1	0	0	0	1	1	0	1	1	1
1	0	0	0	0	1	1	0	1	0	1	1	0
1	0	0	1	0	1	1	0	1	0	1	1	1
1	0	1	0	0	0	0	0	1	1	0	1	0
1	0	1	1	0	0	0	0	1	1	0	1	1
1	1	0	0	1	1	1	0	1	0	1	1	0
1	1	0	1	1	1	1	1	1	0	1	1	1
1	1	1	0	1	0	1	0	1	1	0	0	1
1	1	1	1	1	0	1	1	1	1	0	0	0

Двоичный код в последнем столбце можно разбить на четыре кодовых слова длины 4 числа следующим образом: 1001, 1101, 0101, 0110.

Таким образом получаем следующий шестнадцатеричный код: $9D56_{16}$ или $9(15)56_{16}$

Ответ: $9D56_{16}$ или $9(15)56_{16}$



Задача 3 (15 баллов)

Циклический возврат камеры наблюдения

Исполнитель Камера используется для управления камерой наблюдения, которая может вращаться ограниченно по вертикали и без ограничений по горизонтали. Исполнитель Камера может выполнять следующие команды:

1) Повернуть на (a, b) , где a, b – целые числа. Эта команда поворачивает камеру на величину вверх по вертикали на угол a , заданный в градусах, и поворачивает его против часовой стрелки на угол b , заданный в градусах. Например, команда Повернуть на $(-2, -30)$ опустит камеру на 2 градуса по вертикали вниз и повернет его на угол 30 по горизонтали по часовой стрелке.

2) Цикл:

ПОВТОРИ n

последовательность команд

КОНЕЦ ПОВТОРИ

Этот цикл означает, что последовательность команд будет выполняться n раз.

Исполнителю Камера был дан для исполнения следующий алгоритм (буквами a, b и n обозначены неизвестные целые числа):

НАЧАЛО

повернуть на $(40, 20)$

ПОВТОРИ n

повернуть на (a, b)

повернуть на $(-3, 60)$

КОНЕЦ ПОВТОРИ

повернуть на $(60, -180)$

КОНЕЦ

Чему равно максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов. Чему равно минимально возможное число полных оборотов (360 градусов), которое при этом совершит камера?

Чему равны значения n , если после выполнения алгоритма камера вернулась в то же состояние, что было до начала выполнения алгоритма, совершив 5 полных оборотов? Чему, при этом равны параметры a и b ?

Решение

Значение параметров исполнителя Камера в результате выполнения алгоритма рассчитывается по формуле:

$$l - l_0 = 40 + n(a - 3) + 60$$

$$f - f_0 = 60 + n(b + 60) - 180$$



В этих уравнениях n – количество повторений команд ПОВТОРИ, l и l_0 – конечное и начальное значение угла вертикального поворота камеры, f и f_0 – конечное и начальное значение угла горизонтального поворота камеры.

Так как камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, $l - l_0 = 0$, а $f - f_0 = 360k$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Знак $+$ соответствует оборотам против часовой стрелки, а знак $-$ соответствует оборотам по часовой стрелке.

С учетом этих условий, получим систему уравнений:

$$40 + n(a - 3) + 60 = 0$$

$$20 + n(b + 60) - 180 = 360k$$

После преобразований, получим:

$$n(a - 3) = -100$$

$$n(b + 60) = 360k + 160$$

Нужно найти натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число n должно быть одновременно делителем чисел 100 и $360k + 160$, где $k = 0, \pm 1, \pm 2, \pm 3, \dots$ – соответствует числу оборотов, которое совершает камера. Наибольшим общим делителем для этих выражений будет число 100. Оно будет соответствовать максимально возможному значению n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, при условии, что $360k + 160$ будет кратно 100. Минимальное число оборотов равно 1, что соответствует $k = -1$.

При совершении 5 полных оборотов возможны два варианта, в зависимости от направления вращения:

$$n(b + 60) = 360 \cdot 5 + 160 = 1960$$

$$n(b + 60) = 360 \cdot (-5) + 160 = -1640$$

Нужно найти наибольшее натуральное n , при котором эта система уравнений разрешима в целых числах относительно a и b – наибольшее число, которое является делителем чисел 100 и 1960 или 100 и 1640. В обоих случаях наибольший общий делитель $n = 20$. Рассчитав значения a и b , получим:

$$a = -100/20 + 3 = -2$$

$$b = 1960/20 - 60 = 38 \text{ – при оборотах против часовой стрелки}$$

$$b = -1640/20 - 60 = -142 \text{ – при оборотах по часовой стрелке}$$

Остальные значения n представляют собой делители числа 20: 1, 2, 4, 5, 10. Для них расчет значений a и b производится по аналогичным формулам.



Ответ: максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 100, при этом минимально возможное число полных оборотов равно 1. При совершении 5 полных оборотов максимальное $n = 20$, $a = -2$, $b = -142$ при оборотах по часовой стрелке и $b = 38$ при оборотах против часовой стрелки.



Задача 4 (20 баллов)

Геологический шифр

В ходе геологоразведочных работ в Арктической зоне РФ была обнаружена неизвестная науке горная порода с уникальными кристаллическими свойствами. Чтобы сохранить приоритет открытия в тайне от конкурентов, старший геолог Пётр передал для анализа коллеге Марии зашифрованные данные, содержащие предварительное название породы из двух слов.

Мария получила два файла:

- 1) rock_sample.dat – файл с основным зашифрованным отчётом.
- 2) thin_section.png – микрофотография тонкого среза породы, в которую Петром был внедрён секретный ключ методом стеганографии, имитирующим природные включения.

Алгоритм действий для Марии выглядит следующим образом:

1) Извлечь скрытые данные из изображения thin_section.png. Для этого необходимо проанализировать младшие биты цветовых каналов пикселей, что позволит получить битовую последовательность.

2) Эту битовую последовательность необходимо декодировать. В результате она должна превратиться в строку в формате JSON без кавычек, которая и содержит секретные ключи для алгоритма RSA. В данной задаче строка в формате JSON без кавычек выглядит следующим образом: {ключ1:значение1,ключ2: значение2,ключ3:значение3}.

3) Используя полученные ключи, можно приступить к дешифровке основного отчёта, хранящегося в файле rock_sample.dat.

В результате анализа младших битов цветовых каналов пикселей изображения thin_section.png Мария получила следующую строку:

1100000101100111001111010110100001110011111110101011010000011
00111110111000111110110001

Пётр, известный своей любовью к горным треккингам, для защиты информации в файле thin_section.png применяет специальный «Горный код» стеганографии. Его принцип основан на использовании неравномерного кодирования, где частым символам соответствуют короткие коды, что минимизирует общий размер секретного сообщения. Дополнительный уровень защиты обеспечивают «мусорные» символы – специальные ловушки, встроенные в код. При расшифровке легитимный получатель должен распознать и отфильтровать этот мусор, чтобы получить исходные данные. Таблица используемых Петром кодов представлена ниже.

Символ	Код	Символ	Код
{	110000	7	111100
}	110001	8	111101



Символ	Код	Символ	Код
"	110010	9	111110
:	110011	n	000
,	110100	e	001
0	110101	d	010
1	110110	p	0110
2	110111	q	0111
3	111000	x	1000
4	111001	z	1001
5	111010	y	1010
6	111011		

Неравномерный код – это код, в котором кодовые слова, например, последовательности битов, имеют разную длину.

Например, при декодировании битовой последовательности 000001010 с помощью вышепредставленной таблицы, процесс декодирования будет заключаться в следующем:

1) Читаем биты слева-направо: 0 → 00 → 000.

000 есть в таблице? Да, это n.

Результат: n.

2) Продолжаем со следующего бита: 0 → 00 → 001.

001 есть в таблице? Да, это e.

Результат: e.

3) Продолжаем со следующего бита: 0 → 01 → 010.

010 есть в таблице? Да, это d.

Результат: d.

Расшифрованное сообщение n e d.

Файл rock_sample.dat представляет собой последовательность десятичных чисел, каждая из которых – зашифрованная буква передаваемого сообщения. Исходный текст отчёта перед шифрованием был преобразован в числа по правилу: каждый символ кодировался своим порядковым номером в алфавите (a=1, б=2, ..., я=33). Для шифрования Петром использовалась формула: $C_i = M_i^e \bmod n$,

где C_i – i-й элемент последовательности шифротекста C;

M_i – i-й элемент последовательности шифра M;

e – открытая экспонента – параметр открытого ключа RSA – число, используемое при шифровании в качестве степени;

n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.

Для дешифрования должна использоваться формула $M_i = C_i^d \bmod n$,

где M_i – i-й элемент последовательности шифра M;

C_i – i-й элемент последовательности шифротекста C;

d – секретная экспонента – параметр закрытого ключа RSA – число, используемое при дешифровке в качестве степени;



n – модуль операции взятия остатка;

mod – операция взятия остатка от деления.

Совокупность значений (e, n) и (d, n) является открытым и закрытым ключами, используемыми при шифровании и дешифровании соответственно.

Например, RSA-шифрование с открытым ключом: $(e, n) = (17, 3233)$ – и закрытым ключом: $(d, n) = (2753, 3233)$ заключается в следующем.

При шифровании сообщения $M = 65$ отправляемый шифротекст $C = M^e \text{mod } n = 65^{17} \text{mod } 3233 = 2790$

При дешифровке полученного шифротекста $C = 2790$ восстановленное сообщение $M = C^d \text{mod } n = 2790^{2753} \text{mod } 3233 = 65$.

Примечание

Чтобы вычислить выражение вида $M = C^d \text{mod } n$, иногда невыгодно сначала находить большое число C^d , ведь оно может быть огромным. Удобнее использовать алгоритм быстрого возведения в степень по модулю.

Идея алгоритма:

1. Разложить показатель степени на степени двойки.
2. Последовательно возводить число в квадрат.
3. После каждого шага брать остаток по модулю.
4. Умножать только те степени, которые нужны по разложению.

Например, если нужно найти значение выражения $11^5 \text{mod } 9$, то степень 5 можно представить как сумму степеней 4 и 1: $5 = 4 + 1$.

Следовательно:

1. Находим остаток: $11^2 \text{mod } 9 = 4$, т.к. $9 \times 13 = 117$ (остаток 4).

2. Используем найденный остаток, чтобы получить $11^4 \text{mod } 9$.

$11^4 \text{mod } 9 = (11^2)^2 \text{mod } 9 \equiv 4^2 \text{mod } 9 = 7$, т.к. $9 \times 1 = 9$ (остаток 7)

3. Перемножаем найденный остаток с 11 – и снова берём остаток.

$11^5 \text{mod } 9 = (11^4 \times 11) \text{mod } 9 \equiv (7 \times 11) \text{mod } 9 = 77 \text{mod } 9 = 5$, т.к. $9 \times 8 = 72$ (остаток 5)

Таким образом, $11^5 \text{mod } 9 = 5$.

Используя данные, извлечённые из стеганографического сообщения, расшифруйте rock_sample.dat и узнайте предварительное название новой горной породы. Содержимое rock_sample.dat:

Первое слово – 53 61 10 5 12 62 65; второе слово – 44 39 12 13 39 49 15.

В ответе укажите два слова расшифрованного сообщения в формате «Первое слово» «Второе слово» (например, Медное золото).

Решение

На первом этапе необходимо определить секретный ключ, который был внедрен Петром методом стеганографии в файл thin_section.png.

В задаче сказано, что Пётр использовал для шифрования специальный вод, который основан на использовании неравномерного кодирования. Алгоритм дешифровки в данном случае заключается в следующем: читаем строку по битам слева направо, пошагово накапливая последовательность,



пока она не совпадёт с одним из кодов в таблице. Как только совпадение найдено, записываем символ и начинаем формировать следующую последовательность заново.

В условии сказано, что Мария в результате анализа данного изображения получила битовую строку:

11000001011001110011110101101000011100111111010101101000001100111110111000111110110001

Начнём процесс декодирования. Если код не найден сразу (например, при чтении первой 1), то показан процесс накопления последовательности: $1 \rightarrow 11 \rightarrow 110$ и т.д., до момента совпадения с записью в таблице кодов, используемых Петром при шифровании. Максимальная длина кодового слова в таблице кодировки – 6 бит (например, 110000, 110011, 111110 и т.д.), поэтому в процессе накопления битовой последовательности последовательности длиннее 6 символов не рассматриваются.

Шаг	Индекс начала	Биты (извлечённые)	Найденный код	Символ
1	0	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110000$	110000	{
2	6	$0 \rightarrow 01 \rightarrow 010$	010	d
3	9	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
4	15	$1 \rightarrow 10 \rightarrow 100 \rightarrow 1001$	1001	z
5	19	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111010$	111010	5
6	25	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1101 \rightarrow 11010 \rightarrow 110100$	110100	,
7	31	$0 \rightarrow 00 \rightarrow 001$	001	e
8	34	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
9	40	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1111 \rightarrow 11111 \rightarrow 111110$	111110	9
10	46	$1 \rightarrow 10 \rightarrow 101 \rightarrow 1010$	1010	y
11	50	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1101 \rightarrow 11010 \rightarrow 110100$	110100	,
12	56	$0 \rightarrow 00 \rightarrow 000$	000	n
13	62	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11001 \rightarrow 110011$	110011	:
14	65	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1110 \rightarrow 11101 \rightarrow 111011$	111011	6
15	69	$1 \rightarrow 10 \rightarrow 100 \rightarrow 1000$	1000	x
16	75	$1 \rightarrow 11 \rightarrow 111 \rightarrow 1111 \rightarrow 11111 \rightarrow 111110$	111110	9
17	81	$1 \rightarrow 11 \rightarrow 110 \rightarrow 1100 \rightarrow 11000 \rightarrow 110001$	110001	}

Собирая полученные символы в порядке их дешифровки, получаем точную расшифровку битовой строки: {d:z5,e:9y,n:6x9}. В условии задачи говорится про «мусорные символы» – ловушки, которые Пётр вставляет в сообщение. В таблице кодов есть набор букв-«включений» x, y, z (и т.д.), и



они выступают как мусор в конкретном сообщении: они появляются внутри ключей/значений, но не несут смысла для конечного JSON.

Если удалить все мусорные буквы x, y, z из полученной строки {d:z5,e:9y,n:6x9}, то получим корректную JSON-подобную структуру без кавычек: {d:5,e:9,n:69}. Следовательно, получены данные для расшифровки текстового сообщения из файла rock_sample.dat, а именно модуль $n=69$ и секретная экспонента $d=5$.

Файл rock_sample.dat содержит две последовательности целых чисел, где каждое число – зашифрованный символ шифротекста:

Первое слово: 53 61 10 5 12 62 65

Второе слово: 44 39 12 13 39 49 15

Из условия задачи воспользуемся формулой дешифрования: $M_i = C_i^d \bmod n$,

где M_i – i -й элемент последовательности шифра M (восстановленный номер буквы ($1 = a, 2 = б, \dots, 33 = я$));

C_i – i -й элемент последовательности шифротекста C ;

d – секретная экспонента;

n – модуль операции взятия остатка;

$\bmod$ – операция взятия остатка от деления.

Дешифруем первое слово: $C = [53 \ 61 \ 10 \ 5 \ 12 \ 62 \ 65]$. Можно производить дешифровку напрямую возводя числа в степень $d=5$ или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 53$

$$53^2 = 625 \rightarrow 625 \bmod 69 = 49, \text{ т. к. } 69 \times 40 = 2760 \text{ (остаток 49)}$$

$$\text{Таким образом, } 53^2 \equiv 49 \pmod{69}$$

$$53^4 = (53^2)^2 \equiv 49^2 = 2401 \rightarrow 2401 \bmod 69 = 55, \text{ т. к. } 69 \times 34 = 2346 \text{ (остаток 55)}$$

$$\text{Таким образом, } 53^4 \equiv 55 \pmod{69}$$

$$53^5 = 53^4 \times 53 \equiv 55 \times 53 = 2915 \rightarrow 2915 \bmod 69 = 17, \text{ т. к. } 69 \times 42 = 2898 \text{ (остаток 17)}$$

$$\text{Таким образом, } M_1 = 53^5 \pmod{69} = 17$$

Следовательно, $M_1 = 17$, что соответствует букве алфавита «п».

2. $C_2 = 61$

$$61^2 = 3721 \rightarrow 3721 \bmod 69 = 64, \text{ т. к. } 69 \times 53 = 3657 \text{ (остаток 64)}$$

$$\text{Таким образом, } 61^2 \equiv 64 \pmod{69}$$

$$61^4 = (61^2)^2 \equiv 64^2 = 4096 \rightarrow 4096 \bmod 69 = 25, \text{ т. к. } 69 \times 59 = 4071 \text{ (остаток 25)}$$

$$\text{Таким образом, } 61^4 \equiv 25 \pmod{69}$$

$$61^5 = 61^4 \times 61 \equiv 25 \times 61 = 1525 \rightarrow 1525 \bmod 69 = 7, \text{ т. к. } 69 \times 22 = 1518 \text{ (остаток 7)}$$

$$\text{Таким образом, } M_2 = 61^5 \pmod{69} = 7$$

Следовательно, $M_2 = 7$, что соответствует букве алфавита «ё».

3. $C_3 = 10$



$$10^2 = 64 \rightarrow 100 \bmod 69 = 31, \text{ т. к. } 69 \times 1 = 69 \text{ (остаток 31)}$$

Таким образом, $10^2 \equiv 31 \pmod{69}$

$$10^4 = (10^2)^2 \equiv 31^2 = 961 \rightarrow 961 \bmod 69 = 64, \text{ т. к. } 69 \times 13 = 897 \text{ (остаток 64)}$$

Таким образом, $10^4 \equiv 64 \pmod{69}$

$$10^5 = 10^4 \times 10 \equiv 64 \times 10 = 640 \rightarrow 640 \bmod 69 = 19, \text{ т. к. } 69 \times 9 = 621 \text{ (остаток 19)}$$

Таким образом, $M_3 = 10^5 \pmod{69} = 19$

Следовательно, $M_3 = 19$, что соответствует букве алфавита «с».

4. $C_4 = 5$

$$5^5 = 3125 \rightarrow 3125 \bmod 69 = 20, \text{ т. к. } 69 \times 45 = 3105 \text{ (остаток 20)}$$

Таким образом, $M_4 = 5^5 \pmod{69} = 20$

Следовательно, $M_4 = 20$, что соответствует букве алфавита «т».

5. $C_5 = 12$

$$12^2 = 144 \rightarrow 144 \bmod 69 = 3, \text{ т. к. } 69 \times 2 = 138 \text{ (остаток 6)}$$

Таким образом, $12^2 \equiv 6 \pmod{69}$

$$12^4 = (12^2)^2 \equiv 6^2 = 36 \rightarrow 36 \bmod 69 = 36$$

Таким образом, $12^4 \equiv 36 \pmod{69}$

$$12^5 = 12^4 \times 12 \equiv 36 \times 12 = 432 \rightarrow 432 \bmod 69 = 18, \text{ т. к. } 69 \times 6 = 414 \text{ (остаток 18)}$$

Таким образом, $M_5 = 12^5 \pmod{69} = 18$

Следовательно, $M_5 = 18$, что соответствует букве алфавита «р».

6. $C_6 = 62$

$$62^2 = 3844 \rightarrow 3844 \bmod 69 = 49, \text{ т. к. } 69 \times 55 = 3795 \text{ (остаток 49)}$$

Таким образом, $62^2 \equiv 49 \pmod{69}$

$$62^4 = (62^2)^2 \equiv 49^2 = 2401 \rightarrow 2401 \bmod 69 = 55, \text{ т. к. } 69 \times 34 = 2346 \text{ (остаток 55)}$$

Таким образом, $62^4 \equiv 55 \pmod{69}$

$$62^5 = 62^4 \times 62 \equiv 55 \times 62 = 3410 \rightarrow 3410 \bmod 69 = 29, \text{ т. к. } 69 \times 49 = 3381 \text{ (остаток 29)}$$

Таким образом, $M_6 = 62^5 \pmod{69} = 29$

Следовательно, $M_6 = 29$, что соответствует букве алфавита «ы».

7. $C_7 = 65$

$$65^2 = 4225 \rightarrow 4225 \bmod 69 = 16, \text{ т. к. } 69 \times 61 = 4209 \text{ (остаток 16)}$$

Таким образом, $65^2 \equiv 16 \pmod{69}$

$$65^4 = (65^2)^2 \equiv 16^2 = 256 \rightarrow 256 \bmod 69 = 49, \text{ т. к. } 69 \times 3 = 207 \text{ (остаток 49)}$$

Таким образом, $65^4 \equiv 49 \pmod{69}$

$$65^5 = 65^4 \times 65 \equiv 49 \times 65 = 3185 \rightarrow 3185 \bmod 69 = 11, \text{ т. к. } 69 \times 46 = 3174 \text{ (остаток 11)}$$

Таким образом, $M_7 = 65^5 \pmod{69} = 11$

Следовательно, $M_7 = 11$, что соответствует букве алфавита «й».

Соединив расшифрованные буквы, получаем слово «пёстрый».



Дешифруем второе слово: $C = [44 \ 39 \ 12 \ 13 \ 39 \ 49 \ 15]$. Можно производить дешифровку напрямую возводя числа в степень $d=5$ или же воспользоваться быстрым возведением в степень через квадраты.

1. $C_1 = 44$

$$44^2 = 1936 \rightarrow 1936 \bmod 69 = 4, \text{ т. к. } 69 \times 28 = 1932 \text{ (остаток 4)}$$

$$\text{Таким образом, } 44^2 \equiv 4 \pmod{69}$$

$$44^4 = (44^2)^2 \equiv 4^2 = 16 \rightarrow 16 \bmod 69 = 16$$

$$\text{Таким образом, } 44^4 \equiv 16 \pmod{69}$$

$$44^5 = 44^4 \times 44 \equiv 16 \times 44 = 704 \rightarrow 704 \bmod 69 = 14, \text{ т. к. } 69 \times 10 = 690 \text{ (остаток 14)}$$

$$\text{Таким образом, } M_1 = 44^5 \pmod{69} = 14$$

Следовательно, $M_1 = 14$, что соответствует букве алфавита «м».

2. $C_2 = 39$

$$39^2 = 1521 \rightarrow 1521 \bmod 69 = 3, \text{ т. к. } 69 \times 22 = 1518 \text{ (остаток 3)}$$

$$\text{Таким образом, } 39^2 \equiv 3 \pmod{69}$$

$$39^4 = (39^2)^2 \equiv 3^2 = 9 \rightarrow 9 \bmod 69 = 9$$

$$\text{Таким образом, } 39^4 \equiv 9 \pmod{69}$$

$$39^5 = 39^4 \times 39 \equiv 9 \times 39 = 351 \rightarrow 351 \bmod 69 = 6, \text{ т. к. } 69 \times 5 = 345 \text{ (остаток 6)}$$

$$\text{Таким образом, } M_2 = 39^5 \pmod{69} = 6$$

Следовательно, $M_2 = 6$, что соответствует букве алфавита «е».

3. $C_3 = 12$

Аналогично C_5 для первого слова, $M_5 = M_3 = 18$, что соответствует букве алфавита «р».

4. $C_4 = 13$

$$13^2 = 169 \rightarrow 169 \bmod 69 = 31, \text{ т. к. } 69 \times 2 = 138 \text{ (остаток 31)}$$

$$\text{Таким образом, } 13^2 \equiv 31 \pmod{69}$$

$$13^4 = (13^2)^2 \equiv 31^2 = 961 \rightarrow 961 \bmod 69 = 64, \text{ т. к. } 69 \times 13 = 897 \text{ (остаток 64)}$$

$$\text{Таким образом, } 13^4 \equiv 64 \pmod{69}$$

$$13^5 = 13^4 \times 13 \equiv 64 \times 13 = 832 \rightarrow 832 \bmod 69 = 4, \text{ т. к. } 69 \times 12 = 828 \text{ (остаток 4)}$$

$$\text{Таким образом, } M_4 = 13^5 \pmod{69} = 4$$

Следовательно, $M_4 = 4$, что соответствует букве алфавита «г».

5. $C_5 = 39$

Аналогично C_2 , $M_2 = M_5 = 6$, что соответствует букве алфавита «е».

6. $C_6 = 49$

$$49^2 = 2401 \rightarrow 2401 \bmod 69 = 55, \text{ т. к. } 69 \times 34 = 2346 \text{ (остаток 55)}$$

$$\text{Таким образом, } 49^2 \equiv 55 \pmod{69}$$

$$49^4 = (49^2)^2 \equiv 55^2 = 3025 \rightarrow 3025 \bmod 69 = 58, \text{ т. к. } 69 \times 43 = 2967 \text{ (остаток 58)}$$

$$\text{Таким образом, } 49^4 \equiv 58 \pmod{69}$$



$$49^5 = 49^4 \times 49 \equiv 58 \times 49 = 2842 \rightarrow 2842 \bmod 69 = 13, \text{ т. к. } 69 \times 41 = 2829 \text{ (остаток 13)}$$

$$\text{Таким образом, } M_6 = 49^5 \bmod 69 = 13$$

Следовательно, $M_6 = 13$, что соответствует букве алфавита «л».

$$7. \ C_7 = 15$$

$$15^2 = 225 \rightarrow 225 \bmod 69 = 18, \text{ т. к. } 69 \times 3 = 207 \text{ (остаток 18)}$$

$$\text{Таким образом, } 15^2 \equiv 18 \pmod{69}$$

$$15^4 = (15^2)^2 \equiv 18^2 = 324 \rightarrow 324 \bmod 69 = 48, \text{ т. к. } 69 \times 4 = 276 \text{ (остаток 48)}$$

$$\text{Таким образом, } 15^4 \equiv 48 \pmod{69}$$

$$15^5 = 15^4 \times 15 \equiv 48 \times 15 = 720 \rightarrow 720 \bmod 69 = 30, \text{ т. к. } 69 \times 10 = 690 \text{ (остаток 30)}$$

$$\text{Таким образом, } M_7 = 15^5 \bmod 69 = 30$$

Следовательно, $M_7 = 30$, что соответствует букве алфавита «ь».

Соединив расшифрованные буквы, получаем слово «мергель».

В ответе сказано указать два слова расшифрованного сообщения в формате «Первое слово» «Второе слово», соединяя получаем пёстрый мергель.

Ответ: пёстрый мергель

**Задача 5 (20 баллов)****Маршруты в Кибертауне**

В футуристическом городе Кибертауне роботы-курьеры доставляют посылки по сложной системе туннелей. Каждый маршрут робота записывается как последовательность команд управления, разделённых точками зарядки *.

Примеры маршрутов:

AB (2 команды без зарядки);

AB*CD (блок из 2 команд, зарядка, блок из 2 команд);

A*B*C (3 блока по 1 команде с зарядками между ними).

Алфавит команд (система управления) поддерживает ровно **K** различных команд (например: A, B, C, D, E, F — движение в разных направлениях).

Маршрут состоит из одного или нескольких блоков команд, блоки разделяются символом зарядки *, каждый блок — это последовательность из 1 или более команд.

Общая длина записи маршрута $\leq N$ символов (включая команды и разделители *), при этом маршрут **не может начинаться с ***, маршрут **не может заканчиваться на ***, два разделителя ** подряд **запрещены** (между зарядками всегда есть хотя бы одна команда).

Найти количество различных корректных маршрутов.

Формат входных данных

Одна строка с двумя целыми числами: **N** и **K** через пробел, **N** — максимальная длина маршрута (в количестве команд, в том числе зарядки), **K** — количество различных команд ($1 \leq N \leq 5000$, $1 \leq K \leq 26$).

Формат выходных данных

Одно целое число — количество корректных маршрутов.

Примеры

стандартный ввод	стандартный вывод
3 2	18

Пояснение к примеру

Например, две команды: A, B.

Корректные маршруты длины ≤ 3 :

без зарядок длины 1: A, B — 2 маршрута;

без зарядок длины 2: AA, AB, BA, BB — 4 маршрута;

без зарядок длины 3: AAA, AAB, ABA, ABB, BAA, BAB, BBA, BBB — 8 маршрутов;

с одной зарядкой (структура 1*1): A*A, A*B, B*A, B*B — 4 маршрута;



с одной зарядкой (структура 2^* или $*2$ невозможны, т.к. нельзя начинать/заканчивать с $*$).

Итого: 18 маршрутов.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных данных, поясните решение. Оцените сложность предложенного алгоритма.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
5 1	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Смотрите решение в первом варианте.



Задача 6 (30 баллов)

Ящики Грю

В коридоре у известного суперзлодея Грю скопилась целая куча ящиков. Все они пронумерованы и расположены вдоль стены в одну линию друг за другом. Грю уже и не помнит, почему он их так пронумеровал, но на крышках этих ящиков есть самые разные номера, в том числе и повторяющиеся.

Будучи в душе перфекционистом, Грю давно хочет навести порядок в коридоре. Под порядком он понимает тот факт, чтобы ящики были расположены так же вдоль стены коридора в одну линию, но уже в порядке – порядке неубывания номеров. И вот, наконец-то сегодня, он решил осуществить свой суперзлодейский план по упорядочиванию ящиков.

Само собой, для этого у него есть целая армия неугомонных миньонов. Грю знает, что, если он даст работу некоторым из них, а остальных проигнорирует, те на него сильно обидятся. По этой причине Грю, вообще-то не отличающийся стандартным мышлением, решил действовать так: он выделит каждому миньону некоторый отрезок из нескольких подряд идущих ящиков и попросит того отсортировать эти ящики по неубыванию. Если миньону достанется такой отрезок из ящиков, что ему придется хотя бы немного их переставить – то миньон будет доволен, однако, если ему достанется уже упорядоченный отрезок, миньон будет считать, что его зря позвали и снова обидится. Чтобы миньоны не подрались, их отрезки не должны пересекаться.

Грю желает выделить отрезки миньонам так, чтобы после их работы весь ряд ящиков так же стал упорядочен по неубыванию – всё прямо как он и задумывал в своих суперзлодейских мечтах.

Теперь пришло время делить фронт работ, то есть набор ящиков на непересекающиеся отрезки. И тут Грю понял, что его супермозг не может решить такую задачу. Следует помочь ему и сказать какое максимально возможное число миньонов он сможет осчастливить, выделив каждому непустой отрезок, внутри которого ящики пока еще не стоят в нужном порядке, а после того, как их поставить в каждом таком отрезке по неубыванию, то и вся последовательность ящиков станет упорядоченной по неубыванию.

Формат входных данных

В первой строке находится одно целое число n – количество ящиков, которые находятся в коридоре у Грю ($1 \leq n \leq 2 \cdot 10^5$).

Во второй строке через пробел указаны целые номера b_i этих ящиков в том порядке, в котором они стоят вдоль стены ($1 \leq b_i \leq 10^{18}$).



Формат выходных данных

Вывести в ответ одно число – **максимальное** количество миньонов, которых следует привлечь к упорядочиванию ящиков по методу Грю, с условием, что каждому достанется непустой неупорядоченный отрезок из ящиков. После того, как каждый миньон упорядочит свой отрезок, вся последовательность ящиков должна оказаться так же упорядоченной по неубыванию.

Примеры

стандартный ввод	стандартный вывод
5 2 1 4 5 3	2
4 3 3 4 5	0
5 7 1 3 2 2	1
11 3 4 3 2 4 7 8 7 12 12 9	3

Комментарии к примерам

В первом примере следует одному миньону выделить отрезок от первого до второго ящика, а второму миньону от третьего до пятого ящика.

Во втором примере набор ящиков уже упорядочен, и миньонам здесь делать нечего.

В третьем наборе единственный способ – выбрать один отрезок из всех ящиков. Нужен один миньон.

В четвёртом примере нужны три миньона. Первому следует выделить отрезок с первого до шестого (это ящики подписаны 3 4 3 2 4 7), второму с седьмого до восьмого (ящики подписаны 8 7), третьему оставшиеся ящики с девятого по одиннадцатый (на них стоят надписи 12 12 9). После того, как миньоны упорядочат каждый свои ящики, последовательность ящиков станет так же упорядоченной. Существуют и другие способы раздать отрезки в этом примере, однако больше трёх миньонов никогда не понадобится.

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, **обязательно добавив подробные комментарии, которые поясняют ключевые шаги алгоритма и его идею** – это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык.

Приведите результат выполнения программы для заданных исходных



данных. Оцените сложность предложенного алгоритма.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
11 4 6 1 3 5 7 10 6 9 11 8	?

Ваше решение должно быть **аккуратно оформлено, читабельно** и соответствовать стандартам выбранного языка программирования.

Решение

Смотрите решение в первом варианте.

**ОТВЕТЫ
ВАРИАНТ 1**

1) 9, 0, 9, 7, 27, 0, 8, 9, 0, 3

2) $4(11)16_{16}$ или $4B16_{16}$

3) максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 50, при этом минимально возможное число полных оборотов равно 1. При совершении 5 полных оборотов максимальное $n = 10$, $a = -11$, $b = -171$ при оборотах по часовой стрелке и $b = 189$ при оборотах против часовой стрелки.

4) сизый конгломер

5)

стандартный ввод	стандартный вывод
6 1	20

6)

стандартный ввод	стандартный вывод
11 4 1 3 5 2 10 7 6 9 11 8	2

ВАРИАНТ 2

1) 10, 0, 10, 4, 30, 0, 5, 10, 30, 4

2) $AF56_{16}$ или $(10)(15)56_{16}$

3) максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 50, при этом минимально возможное число полных оборотов равно 2. При совершении 5 полных оборотов максимальное $n=10$, $a=0$, $b=-198$ при оборотах по часовой стрелке и $b=162$ при оборотах против часовой стрелки.

4) льдистый кремень

5)

стандартный ввод	стандартный вывод
2 3	12

6)

стандартный ввод	стандартный вывод
11 4 1 6 3 5 10 7 6 9 11 8	2



ВАРИАНТ 3

1) 3, 0, 12, 5, 8, 0, 12, 8, 0, 5

2) $29BA_{16}$ или $29(11)(10)_{16}$

3) максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 100, при этом минимально возможное число полных оборотов равно 2. При совершении 5 полных оборотов максимальное $n = 20$, $a = -7$, $b = -36$ при оборотах по часовой стрелке и $b = 144$ при оборотах против часовой стрелки.

4) знойный эффузив

5)

стандартный ввод	стандартный вывод
2 4	20

6)

стандартный ввод	стандартный вывод
11 4 6 1 3 5 10 7 6 9 11 8	2

ВАРИАНТ 4

1) 7, 0, 12, 3, 14, 0, 12, 7, 0, 4

2) $BD56_{16}$ или $(11)(15)56_{16}$

3) максимально возможное значение n , при котором камера после выполнения алгоритма вернется в то же состояние, что было до начала выполнения алгоритма, совершив при этом целое число оборотов равно 100, при этом минимально возможное число полных оборотов равно 1. При совершении 5 полных оборотов максимальное $n = 20$, $a = -2$, $b = -142$ при оборотах по часовой стрелке и $b = 38$ при оборотах против часовой стрелки.

4) пёстрый мергель

5)

стандартный ввод	стандартный вывод
5 1	12

6)

стандартный ввод	стандартный вывод
11 4 6 1 3 5 7 10 6 9 11 8	2