



10 КЛАСС. 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:A\$6)		
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<B\$6);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.

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, $7_{16}=0111_2$). 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.

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 ?



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:

110000001110011111110110100100001011001111101010011101000001010
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).

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.

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.



10 КЛАСС. 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: = "АВВ".

	A	B	C	D
1		=СУММ(A\$1:A\$6)		
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<B\$6);A4;A\$5)		
8		=ЕСЛИ(И(B4<B5;B6>B7);A1;A2)		
9		=ЕСЛИ(B7-B\$8>0;B\$7;ЕСЛИ(B\$6-B\$5;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.

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, $7_{16}=0111_2$). 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.

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 ?



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 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



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).

**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.

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.



10 КЛАСС. 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:A\$6)		
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.

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, $7_{16}=0111_2$). 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.

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 ?



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:

11000000111001110101111101101000001100111110011001111011101000
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).

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, ABV, BAA, BAV, BBA, BBV — 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.

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.

.



10 КЛАСС. 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: = "АВВ".

	A	B	C	D
1		=СУММ(A\$1:A\$6)		
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<B\$6);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.

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, $7_{16}=0111_2$). 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.

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 ?



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).

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.

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.