

Задача А. Гиперкуб

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Гиперкуб — это обобщение понятия трёхмерного куба на N измерений. Нуль-мерным гиперкубом является точка, одномерным — отрезок, двумерным — квадрат. В общем же случае N -мерный гиперкуб — это правильный N -мерный многогранник, каждая из $2 \cdot N$ граней которого является $(N - 1)$ -мерным гиперкубом. Например, для $N = 2$ квадрат — это правильный многоугольник, каждая из $2 \cdot 2 = 4$ сторон которого — отрезок, то есть одномерный гиперкуб. Отметим, что N -мерный гиперкуб имеет 2^N вершин.

Старшеклассник Петя долго разбирался, что же такое гиперкуб, но наконец понял, как этот объект устроен, и ему настолько понравилось, что он даже придумал свою собственную игру на гиперкубе. Игра заключается в следующем.

Рассмотрим N -мерный единичный гиперкуб. Расположим его таким образом, чтобы одна из вершин находилась в начале координат — точке $(0, 0, \dots, 0)$ в N -мерном пространстве, а для любой из остальных вершин каждая координата равнялась бы нулю или единице. В каждой из 2^N вершин запишем по целому неотрицательному числу. Игрок начинает свой путь в начале координат. За один ход можно переместиться из текущей вершины по любому ребру при условии, что сумма координат новой вершины строго больше суммы координат старой. Игра заканчивается, когда игрок попадает в вершину $(1, 1, \dots, 1)$, имеющую максимальную сумму координат — N . Результат игры — сумма чисел во всех посещённых игроком вершинах. Цель игры — пройти по гиперкубу таким образом, чтобы эта сумма (количество очков за игру) оказалась как можно больше.

Петя довольно быстро понял, что между двумя вершинами гиперкуба A и B ребро есть тогда и только тогда, когда все координаты этих вершин (A_1, A_2, \dots, A_N) и (B_1, B_2, \dots, B_N) совпадают, кроме одной, которая равна нулю у одной из вершин (скажем, A) и единице у другой (B). Поскольку при этом $A_1 + A_2 + \dots + A_N + 1 = B_1 + B_2 + \dots + B_N$, то по такому ребру можно перемещаться из A в B , но не наоборот. Однако, сыграв в свою игру, Петя не может с уверенностью сказать, является ли полученная им сумма максимальной, или можно на данном гиперкубе сыграть по-другому и набрать больше очков.

Напишите программу, которая по данному гиперкубу находит максимальную сумму, которую можно получить, сыграв в эту игру.

Формат входных данных

В первой строке записано число N ($1 \leq N \leq 10$) — размерность гиперкуба. В следующих 2^N строках содержится по одному числу в каждой; в $(k + 2)$ -й строке записано C_k ($0 \leq C_k \leq 1000$) — число в вершине с номером k .

Номер вершины вычисляется так: вершина A с координатами (A_1, A_2, \dots, A_N) имеет номер, равный $A_1 \cdot 2^{N-1} + A_2 \cdot 2^{N-2} + \dots + A_{N-1} \cdot 2 + A_N$, то есть координаты просто интерпретируются как двоичная запись номера вершины. В этой нумерации начальная вершина имеет номер 0, а конечная — номер $2^N - 1$.

Формат выходных данных

Выведите одно число — максимальную сумму, которую можно получить при игре на данном гиперкубе.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	21
1	
2	
3	
4	
5	
6	
7	
8	

Пояснение к примеру

Наш маршрут таков:

- вершина 0 (число 1, координаты $(0, 0, 0)$) — начальная
- вершина 4 (число 5, координаты $(1, 0, 0)$)
- вершина 6 (число 7, координаты $(1, 1, 0)$)
- вершина 7 (число 8, координаты $(1, 1, 1)$) — конечная

Наше количество очков: $1 + 5 + 7 + 8 = 21$.

Любой другой маршрут с соблюдением правил игры даёт меньшее количество очков.

Задача В. Коммивояжёр возвращается!

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 5 секунд
Ограничение по памяти: 512 мегабайт

Коммивояжёр возвращается в систему Альфы Центавра! Население системы с нетерпением ждёт его прибытия — каждый хочет приобрести что-нибудь с далёких планет!

Как обычно, коммивояжёр хочет минимизировать транспортные расходы. Он выбирает начальную планету, прилетает туда на межгалактическом корабле, после чего посещает все остальные планеты системы в порядке, минимизирующем суммарную стоимость посещения, и на другом межгалактическом корабле улетает обратно. Естественно, коммивояжёр не хочет летать ни на какую планету дважды.

Найдите оптимальный маршрут для коммивояжёра. Массы больше не могут ждать!

Формат входных данных

В системе Альфы Центавра n планет. Это число записано в первой строке входных данных ($1 \leq n \leq 19$). Следующие n строк содержат по n чисел каждая: j -е число на i -й из этих строк — стоимость перемещения a_{ij} от i -й планеты до j -й. Числа в каждой строке разделены пробелами. Числа a_{ii} не несут полезной информации. Все числа во входных данных положительны и не превосходят 10^8 .

Формат выходных данных

В первой строке выведите минимальную суммарную стоимость посещения всех планет. Во второй строке выведите n чисел через пробел — номера планет системы в порядке их посещения. Если оптимальных маршрутов несколько, можно вывести любой из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	5
8 1 6	3 1 2
3 5 7	
4 9 2	

Задача С. Отметки на подмножествах

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим множество \mathcal{S} , состоящее из n элементов — натуральных чисел $1, 2, \dots, n$.

Сперва отметим несколько подмножеств \mathcal{S} , а также все подмножества этих подмножеств.

Затем снимем все отметки, если они есть, с нескольких подмножеств \mathcal{S} , а также со всех их подмножеств.

Найдите количество отмеченных подмножеств после всех этих операций.

Формат входных данных

В первой строке входных данных заданы через пробел три целых числа n , x и y . Следующие x строк содержат описания подмножеств, отмеченных на первом шаге, по одному на строке; также были отмечены все подмножества этих подмножеств. Наконец, последние y строк входных данных содержат описания подмножеств, с которых сняли отметки на втором шаге, по одному на строке; также были сняты отметки со всех их подмножеств. Описание каждого подмножества имеет вид $k \ a_1 \ a_2 \ \dots \ a_k$, где k — количество элементов данного подмножества ($0 \leq k \leq n$), а a_i — сами элементы (a_i попарно различны, $1 \leq a_i \leq n$). Элементы могут быть перечислены в любом порядке.

Формат выходных данных

В первой строке выведите одно число — количество отмеченных подмножеств после всех описанных операций.

Ограничения

- $1 \leq n \leq 10$
- $0 \leq x, y \leq 1000$

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 1 1 1 1 0	1
2 0 1 2 2 1	0

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 2 1 2 2 2 3 2 1 3	3

Пояснения к примерам

В первом примере на первом шаге ставится отметка на подмножество $\{1\}$ и на пустое подмножество, на втором шаге с пустого подмножества снимается отметка.

Во втором примере отметок нет.

В третьем примере на первом шаге отмеченными оказываются следующие шесть подмножеств: $\{\}, \{1\}, \{2\}, \{3\}, \{1, 2\}$ и $\{2, 3\}$. На втором шаге снимаются отметки с трёх подмножеств $\{\}, \{1\}$ и $\{3\}$.

Задача D. Отметки на подмножествах 2

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Условие такое же, как у предыдущей задачи.

Ограничения

- $1 \leq n \leq 20$
- $0 \leq x, y \leq 1000$

Задача E. Отметки на подмножествах 3

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Условие такое же, как у предыдущей задачи.

Ограничения

- $1 \leq n \leq 26$
- $0 \leq x, y \leq 1000$

Задача F. Сумма «случайных» чисел

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

На столе лежат n шариков, на каждом шарике написано натуральное число. Валя и Витя используют эти шарики, чтобы получать «случайные» числа. Процедура получения «случайного» числа следующая. Сначала Валя берёт со стола некоторое непустое подмножество шариков; при этом необходимо, чтобы на столе остался хотя бы один шарик. Затем Витя также берёт какое-то непустое подмножество шариков, оставшихся на столе; после этого шариков на столе может не остаться. Наконец, Валя и Витя вычисляют «случайное» число $r = a \bmod b$, где a — это сумма чисел на шариках Вали, а b — сумма чисел на шариках Вити; $a \bmod b$ — это остаток от деления a на b . После этого все шарики возвращаются на стол.

Предположим, что Валя выбрала каждое допустимое подмножество шариков, и для каждого из них Витя выбрал по одному разу все допустимые подмножества оставшихся шариков. Найдите сумму всех «случайных» чисел, которые получились при этом.

Формат входных данных

В первой строке входного файла задано целое число n . Вторая строка содержит n целых чисел s_1, s_2, \dots, s_n через пробел; s_i — это число, написанное на i -м шарике.

Формат выходных данных

В первой строке выходного файла выведите одно число — сумму всех полученных у Вали и Вити «случайных» чисел.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1 1	0
3 1 1 1	3
3 3 1 2	8

Ограничения

- $2 \leq n \leq 10$
- $1 \leq s_i \leq 1000$

Пояснения к примерам

В первом примере у Вали и Вити получаются числа $a = b = 1$, а $1 \bmod 1 = 0$. Во втором примере числа, отличные от нуля, получаются, только когда Валя берёт один любой шарик, а Витя — оба оставшихся.

В третьем примере суммируются следующие числа:

$0 = 3 \bmod 1$	$1 = 3 \bmod 2$	$0 = 3 \bmod (1 + 2)$
$1 = 1 \bmod 3$	$1 = 1 \bmod 2$	$1 = 1 \bmod (3 + 2)$
$2 = 2 \bmod 3$	$0 = 2 \bmod 1$	$2 = 2 \bmod (3 + 1)$
$0 = (3 + 1) \bmod 2$	$0 = (3 + 2) \bmod 1$	$0 = (1 + 2) \bmod 3$

Задача G. Сумма «случайных» чисел 2

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Условие такое же, как у предыдущей задачи.

Ограничения

- $2 \leq n \leq 16$
- $1 \leq s_i \leq 1\,000\,000$

Задача H. Сумма «случайных» чисел 3

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Условие такое же, как у предыдущей задачи.

Ограничения

- $2 \leq n \leq 18$
- $1 \leq s_i \leq 10$
- Сумма всех s_i не превосходит 50.

Задача I. Дорога

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В Древнем государстве Оссия было два города, между которыми была проложена дорога длиной S метров. Через каждый метр стояли столбики, на каждом из которых по некоторому принципу (этот секретный принцип был известен только древним монахам Шамбалы) было написано по букве (а алфавит там у них был латинский).

Однажды князь-король Василий I решил, что человек, когда он едет по этой дороге, слишком редко вспоминает о нём. Он решил это исправить. Для этого он повелел на некоторых столбиках вместо буквы написать «Здесь был Вася». По его представлению, человек, проехав любой участок дороги длиной K метров, должен обязательно хоть раз увидеть такую надпись. Иными словами, среди каждых K идущих подряд столбиков должен оказаться хоть один, на котором буква заменена на надпись. При этом, чтобы не слишком раздражать монахов (а они люди обидчивые), Василий I приказал выбрать для надписи такие столбики, чтобы среди стёртых букв оказалось как можно меньше различных букв латинского алфавита.

Помогите боярам выполнить приказ своего повелителя.

Формат входных данных

В первой строке написано одно целое число K ($1 \leq K \leq 100\,000$). Во второй строке — без пробелов написано S заглавных латинских букв в той последовательности, в которой ими помечены столбики вдоль дороги. Гарантируется, что $K \leq S \leq 100\,000$.

Формат выходных данных

В первой строке выведите N — минимальное количество различных букв латинского алфавита, которые хотя бы на одном столбике придётся стереть, чтобы написать «Здесь был Вася». В следующих N строках выведите те заглавные буквы латинского алфавита, которые потребуются хоть раз стереть. Буквы можно выводить в любом порядке. Если ответов с минимальным N несколько, можно вывести любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 АВА	1 А
2 АВВАА	2 А В

Задача J. Свёртка

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим все подмножества множества $U = \{0, 1, 2, \dots, n - 1\}$. Каждому подмножеству $A = \{a_1, a_2, \dots, a_k\}$ соответствует уникальное целое число, равное $p(A) = \sum_{i=1}^k 2^{a_i}$. Функцию F от n -элементного множества будем задавать массивом целых чисел f длины 2^n так, что значение функции $F(A)$ равно $f[p(A)]$.

Вам даны две функции F и G , нужно найти функцию H такую, что

$$H(A) = \sum_{B \cup C = A} F(B)G(C).$$

Формат входных данных

В первой строке заданы два целых числа n и t ($1 \leq n \leq 16$, $1 \leq t \leq 100$). Здесь n — размер множества U , а t — количество тестовых случаев. Во второй строке заданы целые числа a и b , каждое от 1 до 10^9 . Эти числа используются в следующем генераторе псевдослучайных чисел:

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand16() {
3.     cur = cur * a + b; // вычисляется по модулю 232
4.     return cur / 216; // целое число от 0 до 216 - 1
5. }
```

Тестовые случаи генерируются последовательно. В каждом из них сперва генерируются по порядку элементы массива f (значения функции F), а затем генерируются по порядку элементы массива g (значения функции G). Каждое следующее целое число генерируется вызовом функции `nextRand16()`.

Формат выходных данных

В ответ на каждый тестовый случай выведите в отдельной строке одно целое число: $\left(\sum_A H(A) \cdot (p(A) + 1) \right) \bmod 2^{32}$.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2	2723387430
30 239017	3167905008
16 2	551267264
239 17	1632349120

Пояснения к примерам

Массивы в первом тесте из примера:

f_1 : 3, 113, 3395, 36331, 41370, 61471, 9130, 11774

g_1 : 25547, 45526, 55066, 13590, 14501, 41817, 9356, 18543

h_1 : 76641, 8167827, 273846333, 5284992017,

1656829263, 11450721456, 3699971823, 14260048942

f_2 : 32024, 43238, 51978, 52034, 53714, 38578, 43250, 52338

g_2 : 62834, 50034, 59250, 8050, 44914, 36722, 53106, 20338

h_2 : 2012196016, 6482475400, 8243104152, 15561662464,

7225902008, 16869349792, 22350138288, 44342816072