

Задачи 10-11

Задача 1. count1011: Количество вхождений цифры

В задаче рассмотрим две симметричные системы счисления (далее будем писать сокращённо - ССС): девятиричную ССС и двадцатисемиричную ССС. В девятиричной ССС используются цифры W, X, Y, Z, 0, 1, 2, 3, 4. К записи числа в этой системе приписывают снизу окончание $_{S9}$. $W_{S9}=-4_{10}$, $X_{S9}=-3_{10}$, $Y_{S9}=-2_{10}$, $Z_{S9}=-1_{10}$, $0_{S9}=0_{10}$, $1_{S9}=1_{10}$, $2_{S9}=2_{10}$, $3_{S9}=3_{10}$, $4_{S9}=4_{10}$. Если имеется запись в девятиричной ССС $a_n a_{n-1} \dots a_1 a_0$, то она обозначает число, равное $a_n * 9^n + a_{n-1} * 9^{n-1} + \dots + a_1 * 9 + a_0 * 1$. Например, $623_{10} = 729 - 81 - 27 + 2 = 1 * 9^3 + (-1) * 9^2 + (-3) * 9 + 2 * 1 = 1ZX2_{S9}$. В двадцатисемиричной ССС используются цифры N, O, P, Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D. К записи числа в этой системе приписывают снизу окончание $_{S27}$. $N_{S27}=-13_{10}$, $O_{S27}=-12_{10}$, $P_{S27}=-11_{10}$, $Q_{S27}=-10_{10}$, $R_{S27}=-9_{10}$, $S_{S27}=-8_{10}$, $T_{S27}=-7_{10}$, $U_{S27}=-6_{10}$, $V_{S27}=-5_{10}$, $W_{S27}=-4_{10}$, $X_{S27}=-3_{10}$, $Y_{S27}=-2_{10}$, $Z_{S27}=-1_{10}$, $0_{S27}=0_{10}$, $1_{S27}=1_{10}$, $2_{S27}=2_{10}$, $3_{S27}=3_{10}$, $4_{S27}=4_{10}$, $5_{S27}=5_{10}$, $6_{S27}=6_{10}$, $7_{S27}=7_{10}$, $8_{S27}=8_{10}$, $9_{S27}=9_{10}$, $A_{S27}=10_{10}$, $B_{S27}=11_{10}$, $C_{S27}=12_{10}$, $D_{S27}=13_{10}$. Если имеется запись в двадцатисемиричной ССС $a_n a_{n-1} \dots a_1 a_0$, то она обозначает число, равное $a_n * 27^n + a_{n-1} * 27^{n-1} + \dots + a_1 * 27 + a_0 * 1$. Например, $623_{10} = 729 - 108 + 2 = 1 * 27^2 + (-4) * 27 + 2 * 1 = 1W2_{S27}$. В симметричных системах счисления перед записью числа не ставят ни плюс, ни минус, и когда число положительное, и когда оно отрицательное.

Составьте программу, которая принимает на вход в первой строке цифру J - одну из цифр девятиричной ССС {W, ..., Z, 0, 1, ..., 4}, во второй строке целое число K, записанное в двадцатисемиричной ССС. В этой записи используются десятичные цифры и заглавные латинские буквы {N, ..., Z, 0, 1, ..., 9, A, ..., D}. Длина записи числа K в двадцатисемиричной ССС не более чем 100000. Программа находит, количество вхождений цифры J в запись числа K, если его перевести в девятиричную ССС. В начале записи числа K могут стоять незначащие нули, которые не следует учитывать при подсчёте количества вхождений J=0. Незначащим является любой нуль, стоящий левее первой ненулевой цифры, или, если K = 0, то все нули, кроме самого правого.

Формат входных данных

В первой строке содержится символ J - цифра девятиричной ССС (одна из {W, ..., Z, 0, 1, ..., 4}).

Во второй строке содержится непустая последовательность символов, являющаяся записью в двадцатисемиричной ССС целого числа K (в этой записи не более чем 100000 символов). В записи числа K используются десятичные цифры и заглавные латинские буквы {N, ..., Z, 0, 1, ..., 9, A, ..., D}.

Формат выходных данных

В первой и единственной строке выводится неотрицательное целое число (от 0 до 150000), равное искомому количеству вхождений цифры J в запись числа K в девятиричной ССС.

Примеры

Ввод

4
DD

Вывод

3

Ввод

0
00000000000000000000000000000000

Вывод

1

Ввод

0
20202020202020202020

Вывод

20

Задача 2. vampir1011: Вампиры

В тексте будем использовать девятеричную симметричную систему счисления (сокращённо ССС), описанную в условиях задачи "Количество вхождений цифры". Рассмотрим целые положительные числа специального вида, которые назовём вампиры₉. Каждое число, являющееся вампиром₉, обладает всеми следующими свойствами: 1) длина его записи в девятеричной ССС является чётной (рассматривается запись без незначащих нулей в начале); 2) оно является произведением двух своих клыков -- целых положительных множителей, не равных друг другу, у которых длина записи в девятеричной ССС равна половине длины записи в девятеричной ССС вампира₉; 3) если запись в девятеричной ССС одного из клыков оканчивается нулём, то у другого клыка последняя цифра в записи в девятеричной ССС не может быть нулём; 4) если выписать друг за другом слитно записи в девятеричной ССС клыков, то можно так переставить цифры в образовавшейся записи, что получится запись в девятеричной ССС вампира₉. Например, $1008_{10} = 1340_{9} = 28_{10} * 36_{10} = 31_{9} * 40_{9}$, значит 1008 является вампиром₉. Второй пример, $81648_{10} = 134000_{9} = 252_{10} * 324_{10} = 310_{9} * 400_{9}$, значит 81648 не является вампиром₉, так как нарушено условие о том, что записи обоих клыков в девятеричной ССС не могут заканчиваться нулём. Третий пример, $77616_{10} = 13Y420_{9} = 252_{10} * 308_{10} = 310_{9} * 4Y2_{9}$, значит 77616 является вампиром₉, так как только у одного клыка запись в девятеричной ССС заканчивается нулём.

Составьте программу, которая принимает на вход в первой строке целое положительное число N , а во второй строке последовательность из N целых различных положительных чисел F_i , разделённых пробелами. Число N не более чем 800. Числа F_i не более чем 14300. Программа подсчитывает количество различных чисел вампиров_{S9}, которые можно получить, используя в качестве их клыков числа F_i , и выводит результат подсчёта.

Формат входных данных

В первой строке содержится целое положительное число N — длина последовательности чисел, ($0 < N < 801$).

Во второй строке содержится N различных целых положительных чисел F_i , разделённых пробелами ($0 < F_i < 14301$).

Формат выходных данных

В первой и единственной строке выводится неотрицательное целое число (не меньше чем 0, не больше чем 32000), равное искомому количеству различных чисел вампиров_{S9}, которые можно получить, используя в качестве их клыков числа F_i .

Примеры

Ввод

```
2
28 36
```

Вывод

```
1
```

Ввод

```
10
1 2 3 4 5 6 7 8 9 10
```

Вывод

```
0
```

Ввод

```
5
36 252 324 308 28
```

Вывод

```
2
```

Задача 3. distance: Социальная дистанция

В одном университете озаботились соблюдением социальной дистанции в студенческой столовой. Зал столовой можно представить как прямоугольную сетку размера N по вертикали на M по горизонтали, в каждой клетке которой может сесть посетитель.

В момент времени s_i , когда приходит посетитель администратор узнает момент времени f_i , когда посетитель уйдет и выбирает свободное место исходя из требований:

- место должно быть свободным
- место посетителя максимально удалено от занятых мест в зале,
- если таких мест несколько, то выбираются места с минимальной координатой y по вертикали
- если таких мест несколько, то выбираются места с минимальной координатой x по горизонтали

Расстояние между точками (y_i, x_i) и (y_j, x_j) вычисляется как $L = \text{abs}(x_i - x_j) + \text{abs}(y_i - y_j)$

Вам требуется написать программу, которая для каждого посетителя определяет место, куда ему сесть.

Формат входных данных

В первой строке задаются размеры зала N, M ($1 \leq N, M \leq 100$) и количество посетителей K ($1 \leq K \leq 5000$).

В следующих K строках информация о посетителях s_i, f_i - время прихода и ухода. Времена до 10^9 , гарантируется что все времена различные и $s_i < f_i$. Посетители упорядочены по s_i .

Формат выходных данных

Для каждого посетителя выведите координаты y, x ($1 \leq y \leq N, 1 \leq x \leq M$). Если свободных мест в зале нет, выведите -1 -1 для этого посетителя

Примеры

Ввод

4 4 3

1 3

2 4

5 6

Вывод

1 1

4 4

1 1

Ввод

1 1 2

1 10

2 9

Вывод

1 1

-1 -1

Задача 4. sphere: Разведка

Корпорация СорКосмос для популяризации своей деятельности заказала анимационный фильм о межпланетной космической программе. В нем автоматический зонд “Великий попил” совершил посадку на поверхности Венеры. На борту зонда находится автоматический квадрокоптер “Полный улет”, который будет летать в атмосфере и изучать поверхность планеты.

Квадрокоптер летает по координатной сетке с шагом в одну угловую минуту, так, что его координаты можно представить парой целых чисел (Lat, Lon), где Lat — географическая широта точки, измеряемая в угловых минутах, то есть значение в диапазоне [-5400, 5400], где значению -5400 соответствует южный полюс, а значению 5400 — северный полюс. Lon — географическая долгота точки, измеряемая в угловых минутах, то есть значение в диапазоне (-10800, 10800], где отрицательными числами обозначается западная долгота, а положительными — восточная. Таким образом параллели и меридианы разделяют поверхность планеты на клетки, сторона каждой клетки равна одной угловой минуте.

Для северного и южного полюса географическая долгота может быть произвольным целым числом в диапазоне (-10800, 10800].

Находясь в точке с координатами (Lat, Lon) не на полюсах квадрокоптер изучает поверхность, заключенную в область на сфере, ограниченную дугами координатной сетки между точками (Lat - 1, Lon + 1), (Lat - 1, Lon - 1), (Lat + 1, Lon - 1), (Lat + 1, Lon + 1), то есть покрывает координатный квадрат со стороной в две угловые минуты с центром в точке (Lat, Lon). Мы будем считать, что это — 4 клетки координатной сетки. Находясь на северном полюсе квадрокоптер изучает область севернее широты 5399, а находясь на южном полюсе квадрокоптер изучает всю область южнее широты -5399. Находясь на широтах 5399 и -5399 квадрокоптер изучает и все клетки севернее 5399 и южнее -5399 соответственно.

Программа для квадрокоптера представляет собой последовательность команд ‘F’ — перелететь на следующую точку координатной сетки, ‘L’ — повернуть налево, ‘R’ — повернуть направо, ‘S’ — остановиться. Команда ‘S’ — единственная и последняя в последовательности команд. Команда ‘L’ означает, что если квадрокоптер был ориентирован на север, то после выполнения команды он будет ориентирован на запад, потом на юг, потом на восток. Перед командой ‘F’ может задаваться число повторений этой команды, например, 20 F означает, что квадрокоптер перемещается вперед на 20 угловых минут, исследуя всю область, которую он пролетел. В начальный момент времени квадрокоптер ориентирован на север. Гарантируется, что

в начальный и конечный момент времени квадрокоптер не находится на полюсах, и при пролете полюса квадрокоптер не поворачивает. То есть, при пролете северного полюса квадрокоптер без изменения направления движения меняет северную ориентацию на южную. Пролет северного и южного полюсов в последовательности команд всегда соответствует двум последовательным командам 'F'.

Формат входных данных

На стандартном потоке ввода задаются два целых числа — координаты точки, в которой квадрокоптер начинает свой полет, затем вещественное число — радиус планеты в километрах, затем последовательность команд квадрокоптера, завершающаяся командой S.

Команды и число повторений могут отделяться друг от друга пробельными символами (в том числе переводом строк). Число повторений записывается без пробелов. Число повторений не превосходит 22000. Общая длина перемещения квадрокоптера не превосходит 1000000 клеток.

Формат выходных данных

На стандартный поток вывода напечатайте количество клеток координатной сетки, которые были исследованы квадрокоптером. Вся область севернее широты 5399 учитывается как одна клетка вся область южнее широты -5399 учитывается как одна клетка. Далее выведите вещественное число — площадь исследованной поверхности планеты в километрах. Пролет над клеткой учитывается только один раз, то есть повторный пролет над нею не увеличивает количество исследованных клеток и исследованную площадь.

Площадь исследованной поверхности планеты выводите не менее чем с 10 десятичными знаками (как показано в примере).

Программа, которая правильно вычисляет только количество клеток, но выводит неправильную площадь (например, всегда выводит площадь 0) получит как минимум 50 баллов из 100.

Примеры

Ввод

```
0 0 6000
F 2 F S
```

Вывод

```
10 30.46173639
```

Задача 5. poker1011: Покер

В одном элитном апартаменте было решено провести турнир по спортивному покеру на игровых автоматах. Для проведения турнира на все игровые автоматы загружается

одинаковая последовательность конфигураций, и игроки соревнуются в том, кто выиграет больше партий. Игровые конфигурации обозначаются положительными целыми числами от 1 до 31918903, а конфигурация 0 является специальной конфигурацией, вызывающей перезагрузку автомата. Конфигурация 0 встречается ровно один раз в конце последовательности игровых конфигураций.

Игровой автомат исполняет последовательность конфигураций циклически, то есть дойдя до конфигурации 0 начинает исполнение с начала последовательности конфигураций.

Для загрузки конфигураций в автомат они кодируются следующим образом. Пусть длина последовательности игровых конфигураций равна N . Тогда, добавляя к ним конфигурацию 0, получим последовательность длины $N + 1$. Возьмем $N + 1$ начальных позиций в последовательности (от первой и до $N + 1$) и для каждой начальной позиции возьмем $N + 1$ следующих за ней конфигураций. Получим $N + 1$ последовательностей каждой длины $N + 1$.

Например, если начальная последовательность конфигураций равна:

1 2 3 1

добавляя к ней 0 и беря все возможные начальные позиции, получим 5 последовательностей:

1 2 3 1 0

2 3 1 0 1

3 1 0 1 2

1 0 1 2 3

0 1 2 3 1

отсортируем эти последовательности лексикографически:

0 1 2 3 1

1 0 1 2 3

1 2 3 1 0

2 3 1 0 1

3 1 0 1 2

Мы получили матрицу из $N + 1$ строк и $N + 1$ столбцов. Теперь возьмем в этой матрице последний столбец: 1 3 0 1 2. Это будет закодированная последовательность конфигураций.

Ваша задача — написать программу, которая декодирует закодированную последовательность конфигураций.

Формат входных данных

На стандартном потоке ввода сначала задается число N ($0 < N < 100000$), задающее длину последовательности, за которой следует $N + 1$ целое число — номера конфигураций. Гарантируется, что конфигурация 0 встречается в последовательности ровно один раз.

Формат выходных данных

На стандартный поток вывода напечатайте N чисел — декодированную последовательность конфигураций. Конфигурацию 0 не печатайте.

Примеры

Ввод

4 1 3 0 1 2

Вывод

1 2 3 1

Задача 6. update: Обновление

Для обновления программного обеспечения в одной p2p сети на центральный узел с номером 0 был загружен апдейт состоящий из K частей по 1 мегабайту. В сети N узлов, соединенные каждый с каждым так, что между парой узлов за одну секунду в каждом направлении может передаваться одна часть размера 1 мегабайт. Каждый узел может одновременно принимать и передавать данные. Когда какая-то часть обновления загружается полностью в узел, эта часть постоянно сохраняется в памяти узла. Каждый узел включается и начинает прием/передачу обновления в момент времени t_i от момента загрузки на центральный узел. Вам требуется определить момент времени, когда все части обновления будут загружены на все узлы сети.

Формат входных данных

В первой строке вводится N и K - число узлов в сети, включая центральный и число частей обновления ($1 \leq N \leq 10^5$, $1 \leq K \leq 10^6$). В следующих строках находятся N чисел - времена включения узлов, начиная с нулевого ($0 \leq t_i \leq 10^6$, $t_0 = 0$)

Формат выходных данных

Выведите одно число - минимальный момент времени, когда на всех узлах сети будут все части обновления

Примеры

Ввод

3 3

0

0

0

Вывод

2

Ввод

2 2

0

1

Вывод

3

Задача 7. blackbox: Черный ящик

Вам на исследование дается некоторая [программа](#) без исходного кода. Программа принимает на вход последовательность целых и дробных числа со знаком в десятичной записи. Например, 392, -1.28281 и т. п. Ввод завершается признаком конца файла (в Unix - Ctrl-D). Программа выводит результат на стандартный поток вывода.

Для некорректных входных данных программа выводит строку `invalid`.

Напишите программу, которая воспроизводит функциональность данной программы, то есть для корректных входных данных будет давать тот же результат, что и исходная программа. Можете предполагать, что на вход вашей программы не будут подаваться входные данные, для которых исходная программа выводит `invalid`.

Вы можете использовать [jslinux](#), на который загрузить программу для эмулятора.

После перехода по ссылке в окне браузера загрузится операционная система Linux и вы получите в окне браузера такой скрин:

```
Loading...
Welcome to JS/Linux (i586)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# █
```



Кнопка загрузки файла в операционную систему находится в левом нижнем углу. Загрузите туда этот [файл](#). После загрузки наберите в командной строке команду "ls -l". Вы должны получить следующее:

```
Loading...
Welcome to JS/Linux (i586)

Use 'vfllogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls -l
total 32
-rw-r--r--  1 root   root      114 Jul  5  2020 bench.py
-rw-r--r--  1 root   root       76 Jul  3  2020 hello.c
-rw-r--r--  1 root   root       22 Jun 26  2020 hello.js
-rw-----  1 1000   root    15208 Mar  9 11:20 prog
-rw-r--r--  1 root   root      151 Jul  5  2020 readme.txt
localhost:~#
```

Выполните команду "chmod 755 prog". Затем можно запускать программу на выполнение с помощью команды "./prog".

```
Loading...

Welcome to JS/Linux (i586)

Use 'vflin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls -l
total 32
-rw-r--r--  1 root    root      114 Jul  5  2020 bench.py
-rw-r--r--  1 root    root       76 Jul  3  2020 hello.c
-rw-r--r--  1 root    root      22 Jun 26  2020 hello.js
-rw-----  1 1000    root    15208 Mar  9 11:20 prog
-rw-r--r--  1 root    root     151 Jul  5  2020 readme.txt
localhost:~# chmod 755 prog
localhost:~# ./prog
392 -1.28281
4267bd3ecb710000
be0a0c2457e3a800
```

Примеры

Ввод

392 -1.28281

Вывод

4267bd3ecb710000

be0a0c2457e3a800