

Задача А: Бомбардировка

Президент Горландии решил уничтожить Флатландию. Для этого было решено сбросить на нее n атомных бомб. Известно, что Флатландия имеет вид многоугольника. Военный министр Горландии выбрал точки, в которые будут сброшены бомбы, и силу каждой из бомб. В результате взрыва бомбы с силой g_i все живое в радиусе g_i от точки сброса оказывается уничтожено.

Однако президенту не понравился план министра. Он сказал, что бомбы не полностью уничтожают территорию Флатландию. Тогда министр решил подготовить новый план. Более того, он подготовил k планов сброса бомб. Но президент не доверяет министру. Он попросил вас проверить, какие из планов приводят к полному уничтожению Флатландии, а какие — нет.



Входные данные

Первая строка содержит число m — количество вершин у многоугольника Флатландии ($3 < m < 100$). Следующие m строк содержат по два целых числа — координаты вершин многоугольника в порядке обхода.

Следующая строка содержит число k ($1 < k < 5$) — количество планов, подготовленных министром. Далее следует k описаний планов. Описание каждого плана состоит из числа n ($0 < n < 100$) — количества сбрасываемых бомб, и n троек целых чисел — координат точки, в которую сбрасывается бомба, и радиуса ее поражения.

Выходные данные

Для каждого плана министра выведите «YES», если следуя этому плану удастся полностью уничтожить Флатландию, и «NO» в противном случае.

Пример входа	Пример выхода
4	YES
0 0	NO
6 0	
6 8	
0 8	
2	
1	
3 4 5	
1	
3 4 4	

Задача В: Интерпретатор

Ваша задача разработка интерпретатора упрощенного Бейсика для начальной части симулятора.

Формальное описание языка:

```
var := 'a' .. 'z'
int := '0' .. '65535'
arg := <var> | <int>
op := '+' | '-' | '*' | '/'
comp := '<' | '=' | '>'
test := <arg> <comp> <arg>
expr := <arg> | <arg> <op> <arg>
команда := 'PRINT' <arg> |
           'GOTO' <int> |
           'END' |
           'IF' <test> 'THEN' <команда> |
<var> = <expr>
```

Программа состоит из $N \geq 1$ команд. Последней командой программы является команда END.

Каждая команда Записывается с начала новой строки. Команды (и строки, их содержащие) занумерованы от 1 до N. Номера команд в Записи программы НЕ ПИШУТСЯ. Гарантируется, что длина команды не превосходит 60 символов, и что программа Закончит свое выполнение на команде END после выполнения не более, чем 10000 команд. Команды исполняются последовательно, начиная с первой (естественно, если нет команды GOTO).

Значения команд:

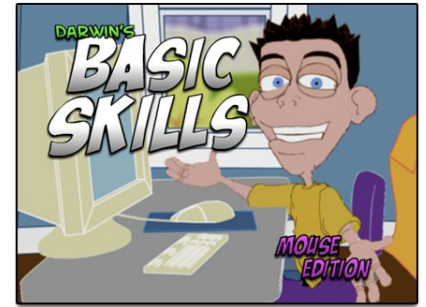
PRINT <arg> - выводит на печать значение константы или переменной, терминированное переводом строки. Гарантируется, что значение переменной, используемой в команде PRINT, определено.

GOTO <int> - перейти к выполнению команд, начиная со строки с заданным номером. Гарантируется, что аргумент команды - целое число от 1 до N.

IF <test> THEN <команда> - если при подстановке значений переменных неравенство или равенство, заданное в условии <test>, становится истинно, то выполняется команда, следующая за THEN.

Иначе выполняется следующая команда. Гарантируется, что значения всех переменных, входящих в <test>, определены.

END - команда, завершающая выполнение программы. При получении этой команды интерпретатор останавливается.



`<var> = <expr>` - Значение выражения `<expr>` вычисляется и присваивается переменной `<var>`.

Гарантируется, что значения всех переменных, входящих в `<expr>`, определены.

Дополнительно:

Все элементы в Записи команд следуют через пробел. Деление производится нацело. Гарантируется, что значение переменной после выполнения арифметической операции не становится меньше нуля или больше 65535. Также гарантируется, что не производится деления на ноль. Гарантируется синтаксическая правильность программы.

Входные данные

В первой строке задана длина `N` программы на упрощенном Бейсике (в командах).

В следующих `N` строках заданы код программы, начиная с первой команды.

Выходные данные

В выходной файл вывести результат выполнения заданной программы на Бейсике (то есть то, что эта программа выведет на печать).

Пример входа	Пример выхода
12	2
p = 2	3
i = 2	5
IF i = p THEN GOTO 9	7
j = p / i	11
j = j * i	13
IF j = p THEN GOTO 10	17
i = i + 1	19
GOTO 3	23
PRINT p	29
p = p + 1	31
IF p < 100 THEN GOTO 2	37
END	41
	43
	47
	53
	59
	61
	67
	71
	73
	79
	83
	89
	97

Задача С: Информатика

В Российско-Армянском университете на одной из практических занятий студентом было задано написать программу на C++ решающую одну математическую задачу.

Одна студентка написала программу даже на двух языках и на Паскале и на C++. Вот ее решения..



C++	Pascal
<pre>#include <iostream> using namespace std; int rec(int N) { if (N == 0 N == 1) return 1; if (N%2 == 0) return (2*rec(N-1) + rec(N-2)) % 101; return (rec(N-1)+rec(N-2)) % 101; } int main() { int N; cin >> N; cout << rec(N) << endl; return 0; }</pre>	<pre>var N:integer; function rec(N:integer):integer; begin if (N = 0) or (N = 1) then begin rec:=1; exit; end; if N mod 2 = 0 then rec:=(2*rec(N-1) + rec(N-2)) mod 101 else rec:=(rec(N-1)+rec(N-2)) mod 101; end; begin read(N); writeln(rec(N)); end.</pre>

Преподаватель одобрил решение, однако сказал, что это не оптимальное решение и не пригодна для больших N. И сейчас он предлагает вам решать ту же задачу, но оптимальнее.

Входные данные

Первая строка содержит одно число N ($0 \leq N \leq 100$).

Выходные данные

Выведите одно число, ответ задачи.

Пример входа	Пример выхода
1	1
11	73

Задача D: Путь через горы

Поверхность Земли в горной местности можно представить в виде ломаной линии. Вершины ломаной расположены в точках (x_1, y_1) , (x_2, y_2) , ..., (x_N, y_N) , при этом $x_i < x_{i+1}$.



Обычный горный маг находится в точке (x_1, y_1) и хочет попасть в точку (x_N, y_N) . При этом он может перемещаться только пешком. Он может ходить по поверхности Земли (т.е. вдоль ломаной). А может сотворить в воздухе мост и пройти по нему. Мост может соединять две вершины ломаной: мост не может начинаться и заканчиваться не в вершине ломаной, и мост не может проходить под землей (в том числе не может быть туннелем в горе), но мост может каким-то своим участком проходить по поверхности земли. Длина моста не может быть больше R . Суммарно маг может построить не более K мостов. Какое наименьшее расстояние придется пройти магу, чтобы оказаться в точке (x_N, y_N) .

Входные данные

Вводится сначала натуральное число N ($2 \leq N \leq 100$). Затем водится натуральное число K ($1 \leq K \leq 100$)— максимальное количество мостов. Далее вводится целое число R ($0 \leq R \leq 10000$)— максимальная возможная длина моста. Далее вводятся координаты (x_1, y_1) , (x_2, y_2) , ..., (x_N, y_N) . Все координаты – целые числа, не превышающие по модулю 10000, для всех i от 1 до $N-1$: $x_i < x_{i+1}$.

Выходные данные

Выведите одно число— минимальную длину пути, которую придется пройти магу (как по земле, так и по мостам). Ответ выведите не менее чем с 5 цифрами после десятичной точки.

Пример входа	Пример выхода
5 2 5 0 0 2 2 3 -1 4 1 5 0	6.47871
9 2 3 1 2 2 1 3 3 5 -1 6 2 7 0 8 1 9 0 10 1	14.93498

Задача E: Контрольная по ударениям



Учительница дала Пете домашнее задание — в заданном тексте расставить ударения в словах, после чего поручила Васе проверить это домашнее задание. Вася очень плохо знаком с данной темой, поэтому он нашел словарь, в котором указано, как ставятся ударения в словах. К сожалению, в этом словаре присутствуют не все слова. Вася решил, что в словах, которых нет в словаре, он будет считать, что Петя поставил ударения правильно, если в этом слове Петей поставлено ровно одно ударение.

Оказалось, что в некоторых словах ударение может быть поставлено больше, чем одним способом. Вася решил, что в этом случае если то, как Петя поставил ударение, соответствует одному из приведенных в словаре вариантов, он будет засчитывать это как правильную расстановку ударения, а если не соответствует, то как ошибку.

Вам дан словарь, которым пользовался Вася и домашнее задание, сданное Петей. Ваша задача — определить количество ошибок, которое в этом задании насчитает Вася.

Входные данные

Вводится сначала число N — количество слов в словаре ($0 \leq N \leq 100$).

Далее идет N строк со словами из словаря. Каждое слово состоит не более чем из 30 символов. Все слова состоят из маленьких и заглавных латинских букв. В каждом слове заглавная ровно одна буква — та, на которую попадает ударение. Слова в словаре расположены в алфавитном порядке. Если есть несколько возможностей расстановки ударения в одном и том же слове, то эти варианты в словаре идут в произвольном порядке.

Далее идет упражнение, выполненное Петей. Упражнение представляет собой строку текста, суммарным объемом не более 30000 символов. Строка состоит из слов, которые разделяются между собой ровно одним пробелом. Длина каждого слова не превышает 30 символов. Все слова состоят из маленьких и заглавных латинских букв (заглавными обозначены те буквы, над которыми Петя поставил ударение). Петя мог по ошибке в каком-то слове поставить более одного ударения или не поставить ударения вовсе.

Выходные данные

Выведите количество ошибок в Петинем тексте, которые найдет Вася.

Пример входа	Пример выхода
4 cAnnot cannOt fOund pAge thE pAge cAnnot be fouNd	2
4 cAnnot cannOt fOund pAge The PAGE cannot be found	4

Задача F: Складирование ноутбуков



На склад, который имеет форму прямоугольного параллелепипеда, привезли ноутбуки, упакованные в коробки. Каждая коробка также имеет форму прямоугольного параллелепипеда.

По правилам хранения коробки с ноутбуками должны быть размещены на складе с выполнением следующих двух условий:

- Стороны коробок должны быть параллельны сторонам склада
- Коробку при помещении на склад разрешается расположить где угодно (с выполнением предыдущего условия), в том числе на другой коробке, но все коробки должны быть ориентированы одинаково (т.е. нельзя одну коробку расположить «стоя», а другую — «лежа»)

Напишите программу, которая по размерам склада и размерам коробки с ноутбуком определит максимальное количество ноутбуков, которое может быть размещено на складе.

Входные данные

Вводится шесть натуральных чисел. Первые три задают длину, высоту и ширину склада. Следующие три задают соответственно длину, высоту и ширину коробки с ноутбуком. Каждое из чисел не превышает 1000.

Выходные данные

Выведите одно число — максимальное количество ноутбуков, которое может быть размещено на складе.

Пример входа	Пример выхода
100 200 300 1 2 3	1000000
100 100 1 2 2 2	0
7 7 7 3 3 3	8
100 200 300 3 2 1	1000000