ВСОШ
ВСЕРОССИЙСКАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Муниципальный этап Всероссийской олимпиады школьников в 2022-2023 учебном году

Предмет	*Класс*	*Дата*	*Время начала*	*Время окончания*
Информатика и ИКТ	***7-8***	***9.12.2022***	***11-00***	***14-00***

Задача 1. Садовник

Максимальная оценка за задачу: `100 первичных баллов`

В этой задаче не нужно сдавать на проверку текст программы. Нужно сдать только ответы на выданные тесты.

Вдоль окон в теплице есть N площадок, на некоторых их них установлены горшки с цветами. Робот ездит вдоль окон от одной стены к другой и считает рядом стоящие горшки и пустые площадки. Его задача переставить горшки так, чтобы они не стояли слишком часто, но чтобы и не было много пустот. Около стены робот разворачивается и едет обратно, к противоположной стене. Он начинает движение от левой стены в направлении к правой. При движении от левой стены он убирает горшки, при обратном движении — ставит, по следующим правилам:

- Если он видит M подряд стоящих горшков (между двумя пустыми площадками, или между пустой площадкой и стеной, стенами) и $K \le M$, то он отъезжает назад на K площадок, убирает один горшок стоящий на этой площадке и продолжает движение в первоначальном направлении. Счет горшков начинается с новой пустой площадки.
- Если он видит, что между двумя горшками (или горшком и стеной) расположено M пустых площадок и $L \le M$, то он отъезжает назад на K площадок, ставит один горшок на эту площадку и продолжает движение к левой стене. Счет пустых площадок начинается с нового горшка.

И так он делает D проездов.

Определите расстановку цветов после полной остановки робота.

Входные данные для одного теста

В первой строке содержится два натуральных числа N и D, где N — количество площадок, D — количество проездов.

Во второй строке содержится два натуральных числа K и L.

В третьей строке дается начальное расположение горшков и пустых площадок в теплице в формате: `Fчисло Eчисло`, где в записи `Fчисло` — число обозначает количество подряд стоящих горшков, в записи `Eчисло` — число обозначает количество подряд стоящих площадок. Перечисление идет в порядке расположения площадок от левой стены к правой. Описание горшков и пустых площадок чередуются.

Выходные данные для одного теста

Ответ должен состоять из одной строки и содержать описание расстановки горшков по окончании работы робота в том же формате, как во входном файле. Описание горшков и пустых площадок обязательно должно чередоваться.

Оценивание

Вам предоставляется 11 входных файлов, названия файлов: `park_01.txt`, `park_02.txt`, ..., `park_11.txt`.

В окно сдачи решения требуется записать ответ из 11-ти строк. В i-й строке запишите ответ для i-го файла, если ответа для этого файла нет, то выведите символ '`?`'. При сдаче ответа укажите компилятор «Текст».

Ответ за каждый файл будет оцениваться отдельно. **Первичная оценка за задачу — сумма баллов за каждый входной файл.**

Пример

Пример ответа содержит ответ на первый файл и пропускает ответы на остальные файлы. Вместо символов '`?`' вставьте свои ответы. Не надо вставлять лишних символов и лишних строк.

Комментарий

Предмет	*Класс*	*Дата*	*Время начала*	*Время окончания*
Информатика и ИКТ	***7-8***	***9.12.2022***	***11-00***	***14-00***

Запись в первом файле `park_01.txt` имеет вид:

```
11 2
4 3
E2 F5 E4
```

ответ

`E2 F1 E1 F3 E2 F1 E1`
`?`
`?`
`?`
`?`
`?`
`?`
`?`
`?`
`?`
`?`

Ответ для этого теста выведен в первой строке примера ответа на задачу. В при первом проезде в направлении слева направо робот убирает один горшок, при втором проезде в обратном направлении — ставит один горшок. **Оценка за ответ на этот файл – 0 баллов.**

Задача 2. Дорога

Входные данные:	`стандартный ввод`
Выходные данные:	`стандартный вывод`
Ограничение по памяти:	`256 МБ`
Ограничение по времени:	`1 секунда на тест`
Максимальная оценка за задачу:	`100 баллов`

Пешеход идет по дороге, перешагивая с одной плитки на следующую. Он знает, что на дороге есть два места, где вместо плиток – глубокие ямы. Пешеход один раз может сделать прыжок фиксированной длины *d*, чтобы попытаться перепрыгнуть через ямы. А дальше опять идет мелкими шажками.

Пешеходу стало интересно узнать, сколько есть способов добраться из начальной точки дороги в конечную. Если в результате прыжка пешеход перепрыгнет конечную плитку, то считается, что он добрался до конечной точки. Способы отличаются последовательностью шагов и прыжка через ямы. Прыжок через последовательность только целых плиток пешеход делать не будет.

Ответом на эту задачу является некоторое выражение. Это выражение должно вычислять в итоге целое число — количество способов добраться из начальной точки дорожки в конечную без падений в ямы. Выражение может содержать целые числа, переменные *N*, *d*, *a*, *b*, операции сложения (обозначается «+»), вычитания (обозначается «−»), умножения (обозначается «*»), деления (обозначается «/»), деления нацело (обозначается «*div*» в Pascal, «//» в Python) и круглые скобки для изменения порядка действий. Запись вида «2*N*» для обозначения произведения числа 2 и переменной *N* неверная, нужно писать «2 * *N*». Пример правильного по форме записи выражения: 10 + (*N*-2) * 2.

Запишите выражение в текст программы на одном из языков программирования. Можете использовать примеры текстов, представленные ниже. Сдайте свой текст программы в окно сдачи решения, указав выбранный компилятор. Во время тура будет только проверка, что выражение записано по описанным правилам.

Входные данные

В первой строке записано одно целое число N ($0 < N \le 100$), N — длина дороги в плитках. Номер первой плитки – 1, последней – N.

Предмет	*Класс*	*Дата*	*Время начала*	*Время окончания*
Информатика и ИКТ	***7-8***	***9.12.2022***	***11-00***	***14-00***

Во второй строке записано одно целое число d ($0 < d \leq 100$), d — длина прыжка, через сколько плиток можно перепрыгнуть. (Если пешеход с плитки с номером 1 сделает прыжок, то он приземлится на плитку с номером $d+2$)

В третьей и четвертой строках записаны номера плиток a и b ($a < b$), вместо которых находятся ямы. Известно, что ямы не находятся на первой и последней плитках.

Выходные данные

Необходимо вывести целое число M — количество способов добраться из начальной точки дороги в конечную.

Язык программирования C++
Компилятор Visual C++ 2019

```
#include<iostream>

using namespace std;
int main()
{
  int N;
  int d, a, b;
  int M;
  cin >> N;
  cin >> d;
  cin >> a;
  cin >> b;

  if(2) /*вместо 2 вставьте сравнение, если
  необходимо */
  M = 1; /*вместо 1 вставьте ответ */
  else
  M=1;
  cout << M << endl;
  return 0;
}
```

Язык программирования Python 3.7.6

```
N = int(input())
d = int(input())
a = int(input())
b = int(input())

if (2): # вместо 2 вставьте сравнение,
#если необходимо
   M = 1 #вместо 1 вставьте ответ
else:
   M = 1 #вместо 1 вставьте ответ

print(M)
```

Язык программирования Pascal
Компилятор PascalABC.NET 3.7.1

```
var
N, d, a, b, M:integer;

begin
  readln(N);
  readln(d);
  readln(a);
  readln(b);
if(True) //вместо true вставьте
      //сравнение, если необходимо
 then
  M := 1 //вместо 1 вставьте ответ
 else
  M := 1; //вместо 1 вставьте ответ

  writeln(M);
end.
```

Предмет	*Класс*	*Дата*	*Время начала*	*Время окончания*
Информатика и ИКТ	***7-8***	***9.12.2022***	***11-00***	***14-00***

Задача 3. Почти анаграмма

Входные данные: стандартный ввод
Выходные данные: стандартный вывод
Ограничение по памяти: 256 МБ
Ограничение по времени: 1 секунда на тест
Максимальная оценка за задачу: 100 баллов

Слово является «почти анаграммой» другого слова, если в одном из слов добавить или убрать ровно одну букву, то слова станут анаграммами. Анаграмма – слово, полученное перестановкой букв, из первоначального слова.

Напишите программу, которая проверяет являются ли слова «почти анаграммами».

Входные данные

В первой строке записано первое слово.

Во второй строке записано второе слово.

Слова состоят из трех маленьких букв латинского алфавита – «a», «b», «c». Длина каждого слова не превышает 256 символов.

Выходные данные

В первой строке выведите `YES` или `NO`. Во второй строке, что нужно сделать с первым словом, если ответ `YES`:

«+» – если в первое слово нужно добавить букву;

«–» – если из первого слова нужно убрать букву;

«0» – если два слова уже анаграммы друг друга.

В третьей строке нужно вывести добавляемую или удаляемую букву.

Пример

стандартный ввод	стандартный вывод
ababc aabb	YES – c
bba bbb	NO

Комментарий

В первом тесте из примера можно убрать в первой строке букву «c», и она станет анаграммой второй строки.

Задача 4. Крепость

Входные данные: стандартный ввод
Выходные данные: стандартный вывод
Ограничение по памяти: 256 МБ
Ограничение по времени: 2 секунды на тест
Максимальная оценка за задачу: 100 баллов

Муниципальный этап Всероссийской олимпиады школьников в 2022-2023 учебном году

Предмет	*Класс*	*Дата*	*Время начала*	*Время окончания*
Информатика и ИКТ	***7-8***	***9.12.2022***	***11-00***	***14-00***

Археологи по космическим снимкам нашли следы древних руин. У археологов есть предположение, что на этой территории была крепость. План крепости – прямоугольник наибольшей площади, в вершинах которого стояли башни. Стены крепости должны быть параллельны осям координат.

Напишите программу, которая среди координат руин находит координаты башен крепости.

Входные данные

В первой строке записано одно целое число N ($0 < N \leq 2000$), N — количество обнаруженных руин.

В каждой из следующих N строк содержится по два целых числа, абсциссы и ординаты очередной руины. Все координаты по модулю не превосходят 10^3. В одной точке может находиться не более одной руины.

Выходные данные

Необходимо вывести в четырех строках координаты башен крепости, по два целых числа в каждой строке. Если есть несколько ответов, то выведите любой. Гарантируется, что крепость можно найти.

Пример

стандартный ввод	стандартный вывод
10 6 0 1 1 2 3 4 4 1 4 6 1 2 1 4 1 7 1 4 3	1 1 1 4 4 1 4 4

Оценивание

Решения, корректно работающие для случаев, когда $N \leq 500$, будут набирать не менее 40 баллов.