**Условная инструкция**

Все ранее рассматриваемые программы имели линейную структуру: все инструкции выполнялись последовательно одна за одной, каждая записанная инструкция обязательно выполняется.

Допустим, мы хотим по данному числу x определить его абсолютную величину (модуль). Для этого необходимо нарушить линейную логику программы. Программа должна напечатать значение переменной x, если x>0 или же величину -x в противном случае. Линейная структура программы нарушается: в зависимости от справедливости условия x>0 должна быть выведена одна или другая величина. Соответствующий фрагмент программы на C++ имеет вид:

     double x;  
     cin>>x;  
     if (x>0)  
     {  
         cout<<x;  
     }  
     else  
     {  
         cout<<-x;  
     }  
     return 0;

В этой программе используется условная инструкция if (если). После слова if в обязательных круглых скобках указывается проверяемое условие (x>0). После этого в фигурных скобках идет блок (последовательность) инструкций, который будет выполнен, если условие истинно, в нашем примере это вывод на экран величины x. Затем идет слово else (иначе) и после него блок инструкций, который будет выполнен, если проверяемое условие неверно.

* [**Задача A.** Максимум из двух чисел](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=292)
* [**Задача B.** Високосный год](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=253)
* [**Задача C.** Тестирующая система](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=2960)
* [**Задача D.** Знак числа](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=2959)
* [**Задача E.** Какое из чисел больше?](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=293)
* [**Задача F.** Максимум из трех](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=294)
* **Задача G. Ладья**
* [**Задача H.** Слон](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=255)
* [**Задача I.** Ферзь](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=256)
* [**Задача J.** Король](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=298)
* [**Задача K.** Конь](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=257)
* [**Задача L.** Шоколадка](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=258)
* [**Задача M.** Фишки](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=259)
* [**Задача N.** Уравнение](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=260)
* [**Задача O.** Сложное уравнение](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=261)
* [**Задача P.** Сдача](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=262)
* [**Задача Q.** Мороженое](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=264)
* [**Задача R.** Котлеты](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=265)
* [**Задача S.** Координатные четверти](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=266)
* [**Задача T.** Существует ли треугольник?](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=295)
* [**Задача U.** Количество равных из трех](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=296)
* [**Задача V.** Квадратное уравнение](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=301)
* [**Задача W.** Тип треугольника](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=302)
* [**Задача X.** Коровы](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=303)
* [**Задача Y.** Билеты на метро](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=304)
* [**Задача Z.** Билеты на метро - 2](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=305)
* [**Задача AA.** Координаты соседей](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=1445)
* [**Задача AB.** Апельсины бочками](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=1448)
* [**Задача AC.** Четные и нечетные числа](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=1451)
* [**Задача AD.** Римская система счисления](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=1459)
* [**Задача AE.** Упорядочить три числа](http://informatics.mccme.ru/mod/statements/view3.php?id=276&chapterid=2961)

**Задача №254. Ладья**

Требуется определить, бьет ли ладья, стоящая на клетке с указанными координатами (номер строки и номер столбца), фигуру, стоящую на другой указанной клетке.

**Входные данные**

Вводятся четыре числа: координаты ладьи (два числа) и координаты другой фигуры (два числа), каждое число вводится в отдельной строке. Координаты - целые числа в интервале от 1 до 8.

**Выходные данные**

Требуется вывести слово YES, если ладья сможет побить фигуру за 1 ход и NO - в противном случае.

**Примеры**

**входные данные**

1

1

2

2

**выходные данные**

NO

**входные данные**

1

1

2

1

**выходные данные**

YES

**Задача №255. Слон**

Требуется определить, бьет ли слон, стоящий на клетке с указанными координатами (номер строки и номер столбца), фигуру, стоящую на другой указанной клетке.

**Входные данные**

Вводятся четыре числа: координаты слона и координаты другой фигуры. Координаты - целые числа в интервале от 1 до 8.

**Выходные данные**

Требуется вывести слово YES, если слон способен побить фигуру за 1 ход, в противном случае вывести слово NO

**Примеры**

**входные данные**

1

1

5

5

**выходные данные**

YES

**входные данные**

3

2

2

2

**выходные данные**

NO

**Задача №257. Конь**

Требуется определить, бьет ли конь, стоящий на клетке с указанными координатами (номер строки и номер столбца), фигуру, стоящую на другой указанной клетке.

**Входные данные**

Вводятся четыре числа: координаты коня и координаты другой фигуры. Все координаты - целые числа в интервале от 1 до 8.

**Выходные данные**

Программа должна вывести слово YES, если конь может побить фигуру за 1 ход, в противном случае вывести слово NO.

**Примеры**

**входные данные**

1

1

3

2

**выходные данные**

YES

**входные данные**

1

1

3

3

**выходные данные**

NO

**Задача №258. Шоколадка**

Требуется определить, можно ли от шоколадки размером *n* × *m* долек отломить *k* долек, если разрешается сделать один разлом по прямой между дольками (то есть разломить шоколадку на два прямоугольника).

**Входные данные**

Вводятся 3 числа: *n*, *m* и *k*; *k* не равно *n* × *m*. Гарантируется, что количество долек в шоколадке не превосходит 30000.

**Выходные данные**

Программа должна вывести слово YES, если возможно отломить указанное число долек, в противном случае вывести слово NO.

**Примеры**

**входные данные**

3

2

4

**выходные данные**

YES

**входные данные**

3

2

1

**выходные данные**

NO

**Задача №259. Фишки**

В каждую крайнюю клетку квадратной доски поставили по фишке. Могло ли оказаться, что выставлено ровно *k* фишек? (Например, если доска 2 x 2, то выставлено 4 фишки, а если 6 x 6 - то 20).

**Входные данные**

Вводится одно натуральное число k.

**Выходные данные**

Программа должна вывести слово YES, если существует такой размер доски, на который будет выставлено ровно (не больше, и не меньше) k фишек, в противном случае - вывести слово NO.

**Примеры**

**входные данные**

20

**выходные данные**

YES

**входные данные**

13

**выходные данные**

NO

**Задача №261. Сложное уравнение**

Решить в целых числах уравнение ( *a*x + *b* ) : ( *c*x + *d* ) = 0.

**Входные данные**

Вводятся 4 числа: *a*, *b*, *c* и *d*; *c* и *d* не равны нулю одновременно.

**Выходные данные**

Необходимо вывести все решения, если их число конечно, “NO” (без кавычек), если решений нет, и “INF” (без кавычек), если решений бесконечно много.

**Примеры**

**входные данные**

1

1

2

2

**выходные данные**

NO

**входные данные**

2

-4

7

1

**выходные данные**

2

**Задача №264. Мороженое**

В кафе мороженое продают по три шарика и по пять шариков. Можно ли купить ровно *k* шариков мороженого?

**Входные данные**

Вводится число *k* (целое,положительное)

**Выходные данные**

Программа должна вывести слово YES, если при таких условиях можно набрать ровно *k* шариков (не больше и не меньше), в противном случае - вывести NO.

**Примеры**

**входные данные**

3

**выходные данные**

YES

**входные данные**

1

**выходные данные**

NO

**Задача №265. Котлеты**

На сковородку одновременно можно положить *k* котлет. Каждую котлету нужно с каждой стороны обжаривать *m* минут непрерывно. За какое наименьшее время удастся поджарить с обеих сторон *n* котлет?

**Входные данные**

Вводятся 3 числа: *k*, *m* и *n*. Все числа не превосходят 32000.

**Выходные данные**

Вывести время, за которое все котлеты будут обжарены.

**Примеры**

**входные данные**

1

5

1

**выходные данные**

10

**Задача №266. Координатные четверти**

Даны координаты двух точек на плоскости, требуется определить, лежат ли они в одной координатной четверти или нет (все координаты отличны от нуля).

**Входные данные**

Вводятся 4 числа: координаты первой точки (*x1*, *y1*) и координаты второй точки (*x2*, *y2*).

**Выходные данные**

Программа должна вывести слово YES, если точки находятся в одной координатной четверти, в противном случае вывести слово NO.

**Примеры**

**входные данные**

3

3

5

1

**выходные данные**

YES

**Задача №303. Коровы**

По данному числу *n* закончите фразу "На лугу пасется..." одним из возможных продолжений: "*n* коров", "*n* корова", "*n* коровы", правильно склоняя слово "корова".

**Входные данные**

Дано число *n* (*n*<100).

**Выходные данные**

Программа должна вывести введенное число *n* и одно из слов (на латинице): korov, korova или korovy, например, 1 korova, 2 korovy, 5 korov. Между числом и словом должен стоять ровно один пробел.

**Примеры**

**входные данные**

1

**выходные данные**

1 korova

**Задача №304. Билеты на метро**

Билет на одну поездку в метро стоит 15 рублей, билет на 10 поездок стоит 125 рублей, билет на 60 поездок стоит 440 рублей. Пассажир планирует совершить *n* поездок. Определите, сколько билетов каждого вида он должен приобрести, чтобы суммарное количество оплаченных поездок было не меньше *n*, а общая стоимость приобретенных билетов – минимальна.

**Входные данные**

Дано одно число *n* - количество поездок.

**Выходные данные**

Выведите три целых числа, равные необходимому количеству билетов на 1, на 10, на 60 поездок.

**Примеры**

**входные данные**

129

**выходные данные**

0 1 2

**Задача №1459. Римская система счисления**

Дано число *X*. Требуется перевести это число в римскую систему счисления.

**Входные данные**

Дано число *X* в десятичной системе счисления (1  ≤  *X*  ≤  100).

**Выходные данные**

Выведите *X* в римской системе счисления.

**Примеры**

**входные данные**

4

**выходные данные**

IV

**входные данные**

19

**выходные данные**

XIX

**входные данные**

46

**выходные данные**

XLVI

**Задача №2961. Упорядочить три числа**

Дано три числа, записанный в отдельных строках. Упорядочите их в порядке неубывания. Программа должна считывать три числа a, b, c, затем программа должна менять их значения так, чтобы стали выполнены условия a <= b <= c, затем программа выводит тройку a, b, c.

**Входные данные**

Вводятся три числа, каждое записано в отдельной строке.

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**входные данные**

1

2

1

**выходные данные**

1 1 2