# Lesson 3. Conditions: if, then, else

1. Input, print and numbers (/en/lessons/r

Theory (/en/lessons/if then else conditions/)

Steps (/en/lessons/if then else conditions/steps/1/)

**Problems** 

2. Integer and float numbers (/en/lessons/i

Conditions: if, then,

else (/en/lessons/i

> **₽** Is positive (/en/less

ls odc (/en/less

🖺 Is even (/en/less

Ends on sever (/en/less

Minimum of two numbers (/en/less

Are both odd (/en/less

🖴 At least one Problem **«Is positive** (/en/lessons/if then else conditions/problems/is positive/)»

#### Statement

Given an integer, print "YES" if it's positive and print "NO" otherwise.

Problem «Is odd (/en/lessons/if then else conditions/problems/is odd/)»

#### **Statement**

Given an integer, print "YES" if it's odd and print "NO" otherwise.

Problem **«Is even** (/en/lessons/if then else conditions/problems/is even/)»

#### Statement

Given an integer, print "YES" if it's even and print "NO" otherwise.

odd (/en/less Problem «Ends on seven (/en/lessons/if\_then\_else\_conditions/problems/ends\_on\_seven/)» Exactly one odd (/en/less **Statement** Sign function Given an integer, print "YES" if it's last digit is 7 and print "NO" otherwise. (/en/less Numbers ascendir Problem «Minimum of two numbers order (/en/less (/en/lessons/if then else conditions/problems/minimum/)» **₽** Is three digit **Statement** (/en/less Given two integers, print the smaller value. Minimum of three numbers (/en/less Equal Problem «Are both odd numbers (/en/lessons/if then else conditions/problems/are both odd/)» (/en/less Rook move Statement (/en/less Given two integers, print "YES" if they're both odd and print "NO" otherwise. board black square (/en/less Chess Problem «At least one odd board -(/en/lessons/if\_then\_else\_conditions/problems/at\_least\_one\_odd/)» same

color (/en/less Statement Given two integers, print "YES" if at least one of them is odd and print "NO" otherwise. Distance to closest point (/en/less Problem «Exactly one odd Digits (/en/lessons/if\_then\_else\_conditions/problems/exactly\_one\_odd/)» ascendir order (/en/less **Statement** 🖴 Four-Given two integers, print "YES" if exactly one of them is odd and print "NO" otherwise. digit palindror (/en/less King move Problem «Sign function (/en/less (/en/lessons/if then else conditions/problems/signum/)» Bishop moves (/en/less **Statement** Queen move For the given integer X print 1 if it's positive, -1 if it's negative, or 0 if it's equal to zero. (/en/less Try to use the cascade if-elif-else for it. Index of outlier (/en/less Knight move Problem «Numbers in ascending order (/en/less (/en/lessons/if then else conditions/problems/are 3 ascending/)» Chocolat bar (/en/less

**Statement** 

Given three different integers, print YES if they're given in ascending order, print NO otherwise. Leap year (/en/less ■ Days in month Problem «Is three digit (/en/less (/en/lessons/if then else conditions/problems/is three digit/)» Next day (/en/less Linea **Statement** equation Given an integer, print "YES" if it's a three-digit number and print "NO" otherwise. (/en/less Vertices of rectangle Problem «Minimum of three numbers (/en/less (/en/lessons/if then else conditions/problems/minimum3/)» 🔓 Sort three numbers (/en/less Statement White Given three integers, print the smallest value. pawn move (/en/less Problem «Equal numbers (/en/lessons/if\_then\_else\_conditions/problems/num\_equal/)»

4. For loop with range (/en/lessons/f

Strings (/en/lessons/s

6. While loop (/en/lessons/v

#### **Statement**

Given three integers, determine how many of them are equal to each other. The program must print one of these numbers: 3 (if all are the same), 2 (if two of them are equal to each other and the third is different) or 0 (if all numbers are different).

7. Lists (/en/lessons/l

8. Functions and recursion (/en/lessons/f

9. Twodimensional lists (arrays) (/en/lessons/t

10. Sets (/en/lessons/s

11.
Dictionaries
(/en/lessons/c

12. JavaScript (/en/lessons/j

13. HTML5 and CSS (/en/lessons/r and-css/)

14.
Responsive
Design
with
Bootstrap
(/en/lessons/k

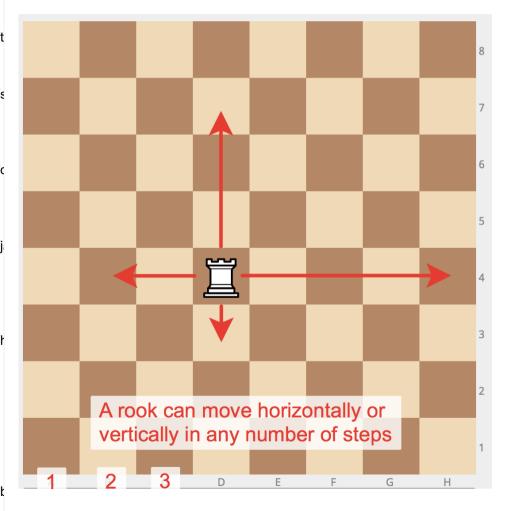
15. jQuery (/en/lessons/j

## Problem «Rook move (/en/lessons/if\_then\_else\_conditions/problems/rook\_move/)»

### **Statement**

Chess rook moves horizontally or vertically. Given two different cells of the chessboard, determine whether a rook can go from the first cell to the second in one move.

The program receives the input of four numbers from 1 to 8, each specifying the column and row number, first two - for the first cell, and then the last two - for the second cell. The program should output YES if a rook can go from the first cell to the second in one move, or NO otherwise.

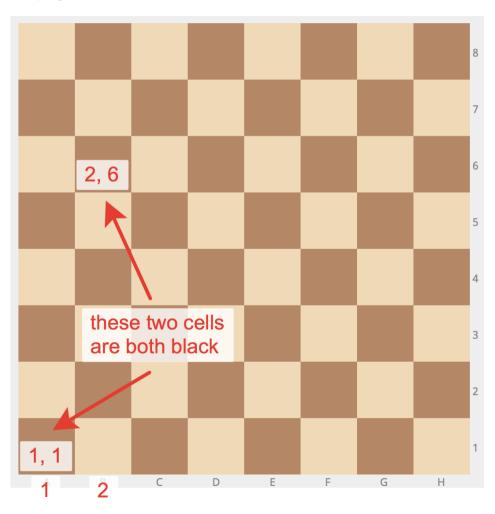


# Problem «Chess board - black square (/en/lessons/if\_then\_else\_conditions/problems/chess\_board\_black\_color/)»

### **Statement**

Given a square of a chessboard. Print BLACK if it's black and print WHITE otherwise.

The program receives two numbers from 1 to 8 each - the column and the row number of the square.





Limited time offer: Get 10 free Adobe Stock images (https: segmei

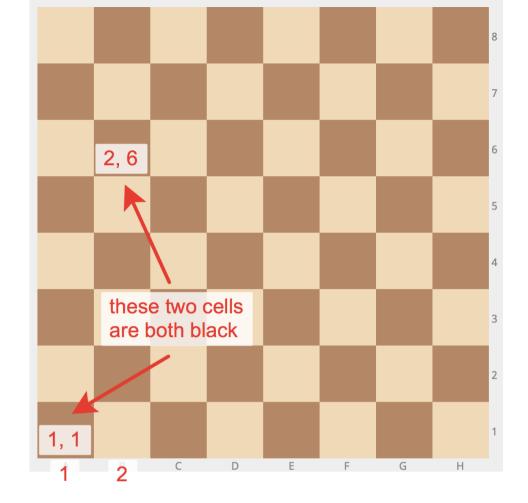
ADS VIA CARBON (HTTP:// UTM\_SOI

# Problem «Chess board - same color (/en/lessons/if\_then\_else\_conditions/problems/chess\_board/)»

#### **Statement**

Given two cells of a chessboard. If they are painted in one color, print the word YES, and if in a different color - NO.

The program receives the input of four numbers from 1 to 8, each specifying the column and row number, first two - for the first cell, and then the last two - for the second cell.



Problem «Distance to closest point (/en/lessons/if\_then\_else\_conditions/problems/distance\_to\_closest/)»

Given the coordinates of the three points A, B, and C on a line. Print a distance from the point A to closest point to it.

### Problem «Digits in ascending order (/en/lessons/if\_then\_else\_conditions/problems/are\_digits\_ascending/)»

#### **Statement**

Given a three-digit integer, print YES if its digits go in ascending order, print NO otherwise.

# Problem **«Four-digit palindrome** (/en/lessons/if\_then\_else\_conditions/problems/four\_digit\_palindrome/)»

#### **Statement**

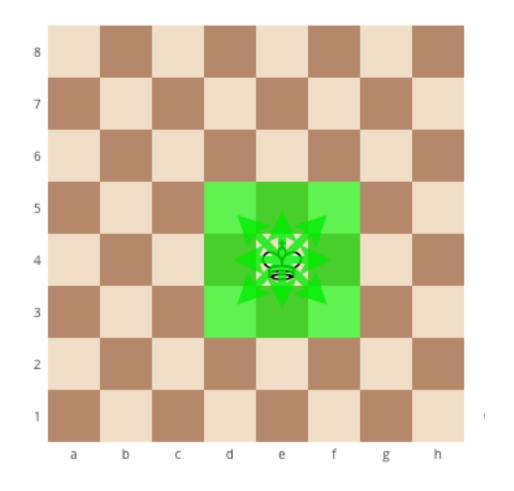
A palindrome is a number which reads the same when read forward as it it does when read backward. Given a four-digit integer, print "YES" if it's a palindrome and print "NO" otherwise.

### Problem «King move (/en/lessons/if\_then\_else\_conditions/problems/king\_move/)»

#### **Statement**

Chess king moves horizontally, vertically or diagonally to any adjacent cell. Given two different cells of the chessboard, determine whether a king can go from the first cell to the second in one move.

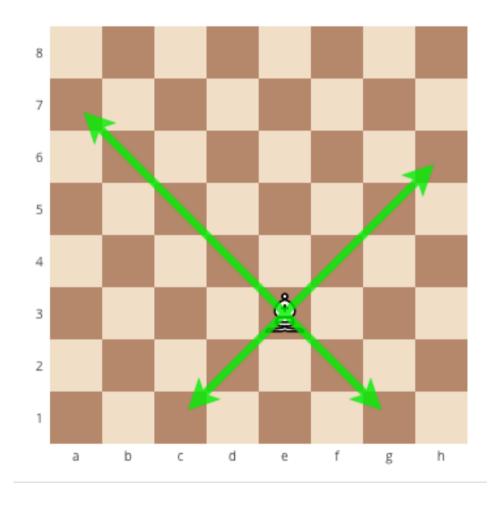
The program receives the input of four numbers from 1 to 8, each specifying the column and row number, first two - for the first cell, and then the last two - for the second cell. The program should output YES if a king can go from the first cell to the second in one move, or NO otherwise.



### Problem «Bishop moves (/en/lessons/if\_then\_else\_conditions/problems/bishop\_move/)»

In chess, the bishop moves diagonally, any number of squares. Given two different squares of the chessboard, determine whether a bishop can go from the first to the second in one move.

The program receives as input four numbers from 1 to 8, specifying the column and row numbers of the starting square and the column and row numbers of the ending square. The program should output YES if a Bishop can go from the first square to the second in one move, or NO otherwise.

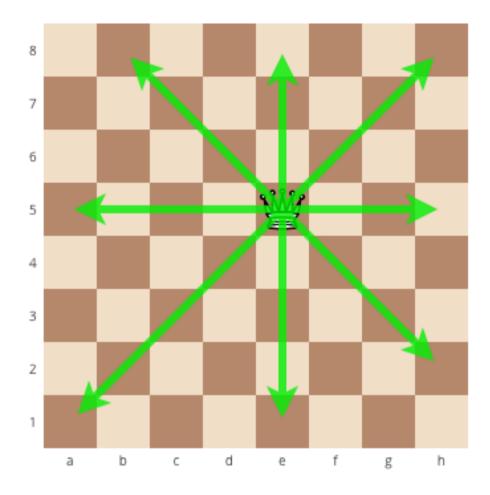


## Problem **«Queen move** (/en/lessons/if\_then\_else\_conditions/problems/queen\_move/)»

#### **Statement**

Chess queen moves horizontally, vertically or diagonally to any number of cells. Given two different cells of the chessboard, determine whether a queen can go from the first cell to the second in one move.

The program receives the input of four numbers from 1 to 8, each specifying the column and row number, first two - for the first cell, and then the last two - for the second cell. The program should output YES if a queen can go from the first cell to the second in one move, or NO otherwise.



Problem «Index of outlier (/en/lessons/if\_then\_else\_conditions/problems/index\_of\_outlier/)»

#### **Statement**

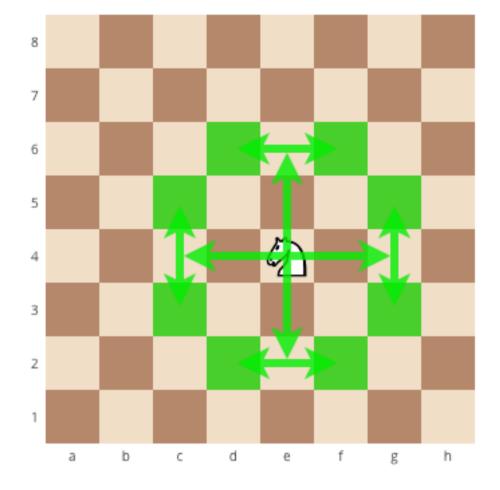
Given three integers: two are equal to each other and the third one is different. Print the index number of this different one - 1, 2 or 3.

### Problem **«Knight move** (/en/lessons/if\_then\_else\_conditions/problems/knight\_move/)»

#### **Statement**

Chess knight moves like the letter L. It can move two cells horizontally and one cell vertically, or two cells vertically and one cells horizontally. Given two different cells of the chessboard, determine whether a knight can go from the first cell to the second in one move.

The program receives the input of four numbers from 1 to 8, each specifying the column and row number, first two - for the first cell, and then the last two - for the second cell. The program should output YES if a knight can go from the first cell to the second in one move, or NO otherwise.



# Problem «Chocolate bar (/en/lessons/if\_then\_else\_conditions/problems/chocolate/)»

### **Statement**

Chocolate bar has the form of a rectangle divided into  $n \times m$  portions. Chocolate bar can be split into two rectangular parts by breaking it along a selected straight line on its pattern. Determine whether it is possible to split it so that one of the parts will have exactly k squares.

The program reads three integers: n, m, and k. It should print YES or NO.

### Problem «Leap year (/en/lessons/if\_then\_else\_conditions/problems/leap\_year/)»

#### **Statement**

Given the year number. You need to check if this year is a leap year. If it is, print LEAP, otherwise print COMMON.

The rules in Gregorian calendar are as follows:

- a year is a leap year if its number is exactly divisible by 4 and is not exactly divisible by 100
- a year is always a leap year if its number is exactly divisible by 400

Warning. The words LEAP and COMMON should be printed all caps.

### Problem «Days in month (/en/lessons/if\_then\_else\_conditions/problems/days\_in\_month/)»

#### **Statement**

Given a month - an integer from 1 (January) to 12 (December), print the number of days in it in the year 2017 (or any other non-leap year).

Problem «Next day (/en/lessons/if\_then\_else\_conditions/problems/next\_day/)»

#### Statement

Given the month (an integer from 1 to 12) and the day in it (an integer from 1 to 31) in the year 2017 (or in any other common year), print the month and the day of the next day to it. The first test corresponds to March 30 and March 31. The second test corresponds to March 31 and April 1.

# Problem «Linear equation (/en/lessons/if\_then\_else\_conditions/problems/linear\_equation/)»

#### Statement

Write a program that solves a linear equation ax = b in integers. Given two integers a and b (a may be zero), print a single integer root if it exists and print "no solution" or "many solutions" otherwise.

# Problem «Vertices of rectangle (/en/lessons/if\_then\_else\_conditions/problems/vertices\_of\_rectangle/)»

#### **Statement**

Given integer coordinates of three vertices of a rectangle whose sides are parallel to the coordinate axes, find the coordinates of the fourth vertex of the rectangle. In the first test the three given vertices are (1, 4), (1, 6), (7, 4). The fourth vertex is thus (7, 6).

## Problem **«Sort three numbers** (/en/lessons/if\_then\_else\_conditions/problems/sort\_three\_numbers/)»

#### **Statement**

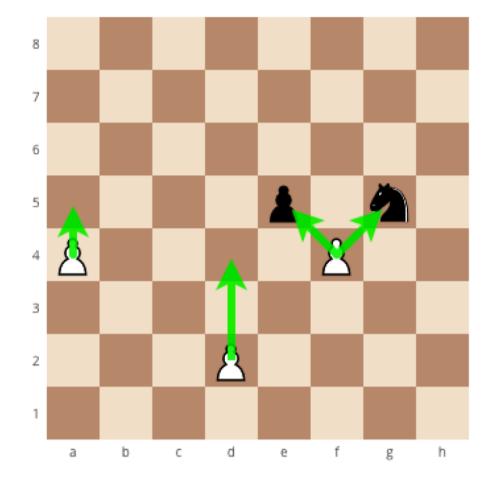
Given three integers, print them in ascending order.

### Problem **«White pawn move** (/en/lessons/if\_then\_else\_conditions/problems/pawn\_move/)»

#### **Statement**

A white chess pawn moves up vertically one square at a time. An exception is a pawn on a row #2: it can move either one or two squares up. In addition, a white chess pawn captures diagonally up one square to the left or right. A white chess pawn can never occur on a row #1.

The program receives the input of four numbers from 1 to 8, each specifying the column and row number, first two - for the first square, and then the last two - for the second square. The program should print YES if a white pawn can possibly move from the first square to the second square in one move in some game - either by move or by capture. The program should print NO otherwise. The first four tests correspond to the green arrows on the picture below.



Maintainer: Vitaly Pavlenko (cxielamiko@gmail.com (mailto:cxielamiko@gmail.com))

Credits to: Denis Kirienko, Daria Kolodzey, Alex Garkoosha, Vlad Sterzhanov, Andrey Tkachev, Tamerlan Tabolov, Anthony Baryshnikov, Denis Kalinochkin, Vanya Klimenko, Vladimir Solomatin, Vladimir Gurovic, Philip Guo

ලී Support us ලී (/support/)

Terms and Conditions (/terms/)
Privacy Policy (/privacy/)

© 2012–2018

Like 1.3K Share

Tweet