|  |  |
| --- | --- |
| 1 | Chess rook walks horizontally or vertically. Two different cells of the chessboard are given, determine whether the rook can get from the first cage to the second one move.  **Input format**  The program receives four numbers from 1 to 8 each, setting the column number and the number of the line first for the first cell, then for the second cell.  **Output format**  The program should withdraw YES if the rook can be hit by the second or NO from the first cell.  **Sample Input:**  4  4  5  5  **Sample Output:**  No |
| 2 | Chess king walks horizontally, vertically and diagonally, but only on one cell. Two different chessboard cells are given, determine whether the king can get from the first cage to the second one move.  **Input format**  The program receives four numbers from 1 to 8 each, setting the column number and the number of the line first for the first cell, then for the second cell.  **Output format**  The program must withdraw YES if the king's move can get into the second or NO from the first cage.  **Sample Input:**  4  4  5  5  **Sample Output:**  YES |
| 3 | Chess elephant walks diagonally. Two different cells of the chessboard are given, determine whether the elephant can get from the first cage to the second one move.  **Input format**  The program receives four numbers from 1 to 8 each, setting the column number and the number of the line first for the first cell, then for the second cell.  **Output format**  The program should withdraw YES if the first cage by the elephant's stroke can get into the second or NO otherwise.  **Sample Input:**  4  4  5  5  **Sample Output:**  YES |
| 4 | The chess queen walks diagonally, horizontally or vertically. Two different cells of the chessboard are given, determine whether the queen can get from the first cage to the second one move.  **Input format**  The program receives four numbers from 1 to 8 each, setting the column number and the number of the line first for the first cell, then for the second cell.  **Output format**  The program must withdraw YES if the first cell by the queen's stroke can get into the second or NO otherwise.  **Sample Input 1:**  1  1  2  2  **Sample Output 1:**  YES  **Sample Input 2:**  1  1  2  3  **Sample Output 2:**  No |
| 5 | Chess horse walks - two cells vertically in any direction and one cell horizontally, or vice versa. Two different cells of the chessboard are given, determine whether the horse can get from the first cage to the second one move.  **Input format**  The program receives four numbers from 1 to 8 each, setting the column number and the number of the line first for the first cell, then for the second cell.  **Output format**  The program should withdraw YES if the first cage by the stroke of the horse can get into the second or NO otherwise.  **Sample Input 1:**  1  1  1  4  **Sample Output 1:**  No  **Sample Input 2:**  1  1  8  8  **Sample Output 2:**  No |
| 6 | The chocolate bar has the appearance of a rectangle, divided into NxM slices. Chocolate can be broken once in a straight line into two parts. Determine whether it is possible to break off from the chocolate in this way exactly K slices.  **Input format**  The program receives three numbers: N, M, K  **Output format**  Программа должна вывести одно из двух слов: YES или NO.  **Sample Input 1:**  4  2  6  **Sample Output 1:**  YES  **Sample Input 2:**  2  10  7  **Sample Output 2:**  No |
| 7 | Yasha swam in a pool the size of the NxM meters and was tired. At this point, he discovered that he was at an X-meter distance from one of the long sides (not necessarily from the nearest one) and the Y meters from one of the short sides. What is the minimum distance Yasha must travel to get out of the pool to the side?  **Input format**  The program receives the entry of the number N, M, X, Y.  **Output format**  The program should bring out the number of meters that you need to swim Yasha to the side.  **Sample Input:**  23  52  8  43  **Sample Output:**  8 |
| 8 | Three numbers are given. Arrange them in non-decreasing order.  **Input format**  Three numbers are given.  **Output format**  Get the answer to the task.  **Sample Input:**  1  2  1  **Sample Output:**  1 1 2 |
| 9 | There are two boxes, A1×B1×C1 and A2×B2×C2 are size of boxes. Define if it is possible to totally cover one box in the another. (Hint: 1x1x1 can be covered by 2x1x1 box)  **Input format**  A1, B1, C1, A2, B2, C2.  **Output format**  The program should bring out one of the following lines:  Boxes are equal, if the boxes are the same,  the first box is smaller than the second one, if the first box can be put in the second,  the first box is larger than the second one, if the second box can be put in the first, Boxes are incomparable, in all other cases.  **Sample Input 1:**  1  2  3  3  2  1  **Sample Output 1:**  Boxes are equal  **Sample Input 2:**  2  2  3  3  2  1 |
| 10 | For this whole number N print out all squares of natural numbers, not exceeding N, in ascending order.  **Input format**  A natural number is introduced.  **Output format**  Get the answer to the task.  **Sample Input:**  50  **Sample Output:**  1 4 9 16 25 36 49 |
| 11 | Given an integer number, not less than 2. Bring out its smallest natural divider, different from the one.  **Input format**  A whole positive number is introduced.  **Output format**  Get the answer to the task.  **Sample Input:**  15  **Sample Output:**  3 |
| 12 | For this number N print out all whole degrees of deuce, not exceeding N, in ascending order.  **Input format**  A natural number is introduced.  **Output format**  Get the answer to the task.  **Sample Input:**  50  **Sample Output:**  1 2 4 8 16 32 |
| 13 | Determine the amount of all the sequence elements that ends in number 0.  **Input format**  A sequence of whole numbers is introduced, ending with the number 0 (the number 0 itself is not included in the sequence, but serves as a sign of its end.  **Output format**  Get the answer to the task.  **Sample Input:**  3  6  8  0  **Sample Output:**  17 |
| 14 | The sequence consists of natural numbers and ends with the number 0. Determine how many elements of this sequence are equal to the largest element of this sequence.  **Input format**  A non-empty sequence of whole numbers is introduced, ending with the number 0 (the number 0 itself is not included in the sequence, but serves as a sign of its end).  **Output format**  Get the answer to the task.  **Sample Input 1:**  1  7  9  0  **Sample Output 1:**  1  **Sample Input 2:**  1  3  3  1  0  **Sample Output 2:**  2 |
| 15 | Fibonacci's sequence is defined as:  F(0) = 0, F(1) = 1, ..., F(n) = F (n-1) + F (n-2).  Given the natural number A. Determine what the number of Fibonacci is, that is, put out such a number N, that F(N) is A. If A is not the number Fibonacci, enter the number -1.  **Input format**  Introduced the natural number A > 1.  **Output format**  Get the answer to the task.  **Sample Input:**  8  **Sample Output:**  6 |
| 16 | The sequence element is called a local maximum if it is strictly larger than the previous and next sequence element. The first and last element of the sequence are not local maximums. The sequence of natural numbers, which ends with the number 0, is given. Determine the number of strict local highs in this sequence.  **Input format**  A sequence of natural numbers is introduced, ending with the number 0 (the number 0 itself is not included in the sequence, but serves as a sign of its end).  **Output format**  Get the answer to the task.  **Sample Input:**  1  2  1  2  1  0  **Sample Output:**  2 |
| 17 | A list is given, orderly by the non-descending of the elements in it.  **Input format**  The first line introduces the number of items in the array. The second line introduces elements of the array.  **Output format**  Get the answer to the task.  **Sample Input:**  6  1 2 2 3 3 3  **Sample Output:**  3 |
| 18 | Cycle the items on the list to the right (A[0]) moves to place A[1], A[1] to place A[2], ... the last element goes to place A[0].  **Input format**  The first line introduces the number of items in the array. The second line introduces elements of the array.  **Output format**  Get the answer to the task.  **Sample Input:**  5  1 2 3 4 5  **Sample Output:**  5 1 2 3 4 |
| 19 | An array of numbers has been given. Calculate how many pairs of elements in it are equal. It is believed that any two elements equal form one pair, which must be counted.  **Input format**  The first line introduces the number of items in the array. The second line introduces elements of the array.  **Output format**  Get the answer to the task.  **Sample Input 1:**  5  1 2 3 2 3  **Sample Output 1:**  2  **Sample Input 2:**  5  1 1 1 1 1 |
| 20 | N pins were put in a row, numbering them from left to right numbers from 1 to N. Then on this row threw K balls, with the i-ball knocked down all the skittles with numbers from li to ri inclusive. Define pins that are still on their place.  **Input format**  Program gets N pins and K throws. Then K pairs of numbers li, ri, and 1 ≤ li, ri ≤ N.  **Output format**  The program should deduce a sequence of N characters where the j-y symbol is "I" if j-I pin is left standing, or "." if j-I pin was knocked down.  **Sample Input:**  10 3  8 10  2 5  3 6  **Sample Output:**  I..... I... |