1 Odd or Even

Description

Write a program that reads an integer from the console, uses an expression to check if given integer is odd or even, and prints "even NUMBER" or "odd NUMBER", where you should print the input number's value instead of NUMBER.

Input

- On the single input line you will receive an integer number.

Output

- Output a single line - if the number is even, output even, followed by a whitespace and the value of the number. Otherwise, print odd, again followed by a whitespace and the number's value. See the sample tests.

Constraints

- The input number will always be a valid integer number.
- The input number will always be in the range `[-30, 30]`.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
3	odd 3
2	even 2
-2	even -2

-1	odd -1
0	even 0

2 Moon Gravity

Description

The gravitational field of the Moon is approximately 17% of that on the Earth.

- Write a program that calculates the weight of a man on the moon by a given weight W(as a floating-point number) on the Earth.
- The weight W should be read from the console.

Input

- The input will consist of a single line containing a single floating-point number - the weight W.

Output

- Output a single floating-point value - how much a man with the weight W on Earth will weigh on the Moon. Output all values with exactly 3-digit precision after the floating point.
    - *Hint: You can use the built-in .NET functionality*

Constraints

- The input weight will always be a valid floating-point number, always between 0 and 1000, exclusive.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
86	14.620
74.6	12.682
53.7	9.129

3 Divide by 7 and 5

Description

Write a program that does the following:

- Reads an integer number from the console.
- Stores in a variable if the number can be divided by 7 and 5 without remainder.
- Prints on the console "true NUMBER" if the number is divisible without remainder by 7 and 5. Otherwise prints "false NUMBER". In place of NUMBER print the value of the input number.

Input

- The input will consist of a single integer value.

Output

- The output must always follow the format specified in the description. See the sample tests.

Constraints

- The input will always consist of valid integers and follow the format in the description.

- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
3	false 3
0	true 0
5	false 5
7	false 7
35	true 35
140	true 140

4 Rectangles

Description

Write an expression that calculates rectangle’s area and perimeter by given width and height. The width and height should be read from the console.

Input

- The input will consist of 2 lines:
    - On the first line, you will receive a floating-point number that will represent the width of the rectangle.
    - On the second line, you will receive another floating-point number that will represent the height of the rectangle.

Output

- Output a single line - the rectangle's area and perimeter, separated by a whitespace.
    - Print the area and perimeter values with *exactly 2-digits precision* after the floating point

Constraints

- The width and height will always be valid floating-point numbers.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
2.5 3	7.50 11.00
5 5	25.00 20.00
3 4	12.00 14.00

5 Third digit

Description

Write a program that reads an integer N from the console and prints true if the third digit of N is 7, or "false THIRD_DIGIT", where you should print the third digits of N.

- The counting is done from right to left, meaning 123456's third digit is 4.

Input

- The input will always consist of exactly one line, containing the number N.

Output

- The output should be a single line - print whether the third digit is 7, following the format described above.

Constraints

- N will always be valid non-negative integer number.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
5	false 0
701	true
9703	true
877	false 8
777877	false 8
9999799	true

6 Four digits

Description

Write a program that takes as input a four-digit number in format abcd (e.g. 2011) and performs the following:

1. Calculates the sum of the digits (in our example `2 + 0 + 1 + 1 = 4`) and prints it on the console.
2. Prints on the console the number in reversed order: dcba (in our example 1102) and prints the reversed number.
3. Puts the last digit in the first position: dabc (in our example 1201) and prints the result.
4. Exchanges the second and the third digits: acbd (in our example 2101) and prints the result.

Input

- The input will consist of a single four-digit integer number on a single line.

Output

- Output the result of each action on a separate line, in the same order as they are explained above:
    - meaning the sum comes on the first line, the reversed number on the second, and so on.

Constraints

- The number will always be a valid positive four-digit integer.
- The number will always start with a digit other than 0.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
2011	4 1102 1201 2101

3333	12 3333 3333 3333
9876	30 6789 6987 9786
1230	6 0321 0123 1320

7 Point in a circle

Description

Write a program that reads the coordinates of a point x and y and using an expression checks if given point (x, y) is inside a circle `K({0, 0}, 2)` - the center has coordinates (0, 0) and the circle has radius 2. The program should then print "yes DISTANCE" if the point is inside the circle or "no DISTANCE" if the point is outside the circle.

- In place of DISTANCE print the distance from the beginning of the coordinate system - (0, 0) - to the given point.

Input

- You will receive exactly two lines, the first containing the x coordinate, the second containing the y coordinate.

Output

- Output a single line in the format described above. The distance should be always printed with *2-digit precision* after the floating point.

Constraints

- The numbers x and y will always be valid floating point numbers in the range `(-1000, 1000)`
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
-2 0	yes 2.00
-1 2	no 2.24
1.5 -1	yes 1.80
-1.5 -1.5	no 2.12
100 -30	no 104.40
0 0	yes 0.00
0.2 -0.8	yes 0.82
0.9 -1.93	no 2.13
1 1.655	yes 1.93

0 1	yes 1.00

8 Prime check

Description

Write a program that reads an integer N (which will always be less than 100 or equal) and uses an expression to check if given N is prime (i.e. it is divisible without remainder only to itself and 1).

- *Note: You should check if the number is positive*

Input

- On the only input line you will receive the number N.

Output

- Output true if the number is prime and false otherwise.

Constraints

- N will always be a valid 32-bit integer number, which will be less than 100 or equal.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
2	true

23	true
-3	false
0	false
1	false

9 Trapezoids

Description

Write an expression that calculates trapezoid's area by given sides a and b and height h. The three values should be read from the console in the order shown below. All three value will be floating-point numbers.

Input

- The input will consist of exactly 3 lines:
    i. The side a on the first line.
    ii. The side b on the second line.
    iii. The height h on the third line.

Output

- Output a single line containing a single value - the area of the trapezoid. Output the area with exactly *7-digit precision* after the floating point.

Constraints

- All numbers will always be valid floating-point numbers in the range `[-500, 500]`.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
5 7 12	72.0000000
2 1 33	49.5000000
8.5 4.3 2.7	17.2800000
100 200 300	45000.0000000
0.222 0.333 0.555	0.1540125

10 Point, Circle, Rectangle

Description

Write a program that reads a pair of coordinates x and y and uses an expression to checks for given point (x, y) if it is within the circle `K({1, 1}, 1.5)` and out of the rectangle `R(top=1, left=-1, width=6, height=2)`.

Input

- You will receive the pair of coordinates on the two lines of the input - on the first line you will find x, and on the second - y.

Output

- Print inside circle if the point is inside the circle and outside circle if it's outside. Then print a single whitespace followed by inside rectangle if the point is inside the rectangle and outside rectangle otherwise. See the sample tests for a visual description.

Constraints

- The coordinates x and y will always be valid floating-point numbers in the range `[-1000, 1000]`.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
2.5 2	outside circle outside rectangle
0 1	inside circle inside rectangle
2.5 1	inside circle inside rectangle
1 2	inside circle outside rectangle

11 3rd Bit

Description

Using bitwise operators, write a program that uses an expression to find the value of the bit at index 3 of an unsigned integer read from the console.

- The bits are counted from right to left, starting from bit 0.
- The result of the expression should be either 1 or 0. Print it on the console.

Input

- On the only input line, you will receive a single positive integer - the number whose 3rd bit you must print.

Output

- Output a single number - 1 or 0 - the value of the 3rd bit, counted from 0 and from right to left.

Constraints

- The input number will always be a valid positive integer number.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Output
15	1
1024	0

12 N-th Bit

Description

Write a program that reads from the console two integer numbers P and N and prints on the console the value of P's N-th bit.

Input

- On the first line you will receive the number P. On the second line you will receive the number N.

Output

- Output a single value - the value of the N-th bit in P.

Constraints

- N will be a positive integer and always smaller than 55.
- 0 <= P <= 2^{55}
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Binary representation	Output
5 2	00000000 00000000 00000101	0
0 9	00000000 00000000 00000000	0
15 1	00000000 00000000 00001111	1
5343 7	00000000 00010100 11011111	1
62241 11	00000000 11110011 00100001	0

985276 49	00001111 00001000 10111100	0

13 Modify Bit

Description

We are given an integer number N (read from the console), a bit value v (read from the console as well) (v = 0 or 1) and a position P (read from the console). Write a sequence of operators (a few lines of C# code) that modifies N to hold the value v at the position P from the binary representation of N while preserving all other bits in N. Print the resulting number on the console.

Input

- The input will consist of exactly 3 lines containing the following:
    i. First line - the integer number N.
    ii. Second line - the position P.
    iii. Third line - the bit value v.

Output

- Output a single line containing the value of the number N with the modified bit.

Constraints

- N will always be a valid 64-bit unsigned integer.
- P will always be between in the range `[0, 64)`.
- v will be always either 0 or 1.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Binary representation	Modified value	Output
5 2 0	0 0000 00000101	00000000 00000001	1
0 9 1	00000000 00000000	00000010 00000000	512
15 1 1	00000000 00001111	00000000 00001111	15
5343 7 0	00010100 11011111	00010100 01011111	5215
62241 11 0	11110011 00100001	11110011 00100001	62241

14 BitSwap

Description

Write a program that first reads 4 numbers n, p, q and k and then swaps bits `{p, p+1, …, p+k-1}` with bits `{q, q+1, …, q+k-1}` of n. Print the resulting integer on the console.

Input

- On the only four lines of the input you will receive the integers n, p, q and k in this order.

Output

- Output a single value - the value of n after the bit swaps.

Constraints

- The first and the second sequence of bits will never overlap.
- n will always be a valid 32-bit positive integer.
- Time limit: 0.1s
- Memory limit: 16MB

Sample tests

Input	Binary representation	Binary representation after swaps	Output
1140867093 3 24 3	01000100 00000000 01000000 00010101	01000010 00000000 01000000 00100101	1107312677
4294901775 24 3 3	11111111 11111111 00000000 00001111	11111001 11111111 00000000 00111111	419423852 7
2369124121 2 22 10	10001101 00110101 11110111 00011001	01110001 10110101 11111000 11010001	1907751121