**WEEK-3**

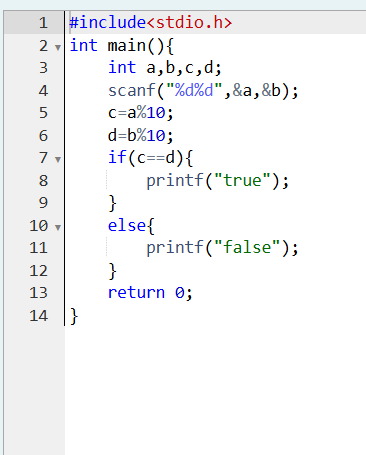
1.Write a program to read two integer values and print true if both the numbers end with the same digit, otherwise print false. Example: If 698 and 768 are given, program should print true as they both end with 8. Sample Input 125 53 Sample Output 1 false Sample Input 2.27.77

Mabert out of

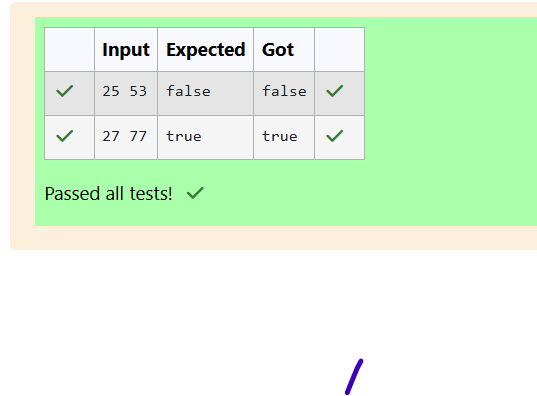
9

Sample Output 2

True



**OUTPUT:**



**2.Objective**

In this challenge, we're getting started with conditional statements.

Task

Given an integer, a, perform the following conditional actions:

If n is odd, print Weird

It n is even and in the inclusive range of 2 to 5, print Not Weird

If n is even and in the inclusive range of 6 to 20, print Weird

Il is even and greater than 20, print Not Weird

Complete the stub code provided in your editor to print whether of not a is weird.

Input Format

D

A single line containing a positive integer, n

Constraints

1<=n<=10

Output Format

Print Weird it the number is werd otherwise, print Not Weind

Sample Input O

Sample Output 0

Weerd

Sample Input 1

74

Sample Output 1

Not Weird

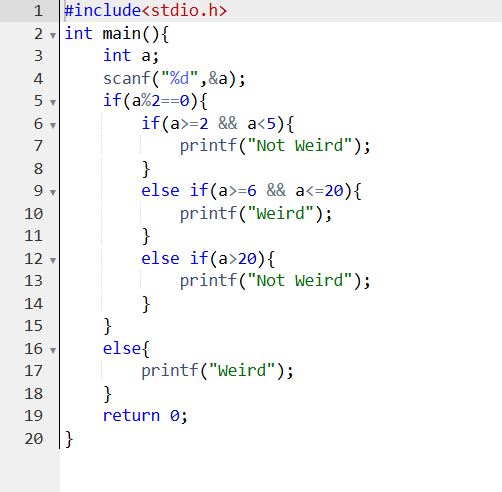
Explanation

Sample Case 0: n=3

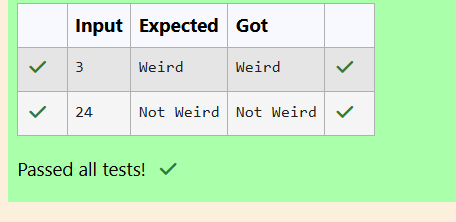
n is odd and odd numbers are weird, so we print Weird.

Sample Case 1: n = 24

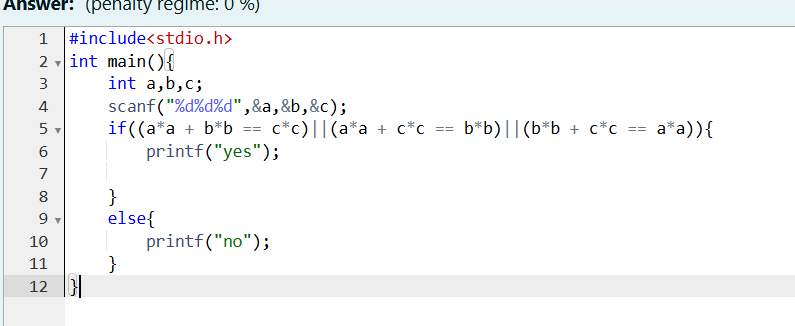
n > 20 and n is even, so it isn't weird. Thus, we print Not Weird



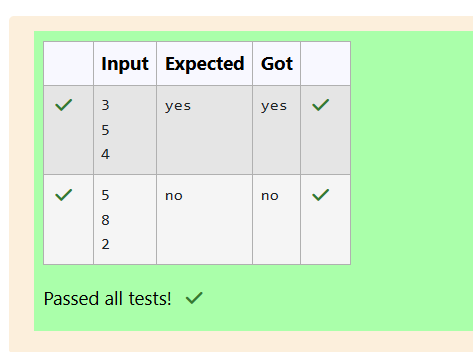
**OUTPUT:**



1. Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third. For example, 3. 5 and 4 form a Pythagorean triple, since 3\*3 + 4\*4 = 25 = 5\*5 You are given three integers, a, b, and c. They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters. Sample Input 1 3.5 4 Sample Output 1 yes Sample Input 2 5 8 2 Sample Output 2 no



**OUTPUT:**



**1**.Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to tand including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message

Sample Input 1

Sample Output 1

Triangle

Sample Input 2

Sample Output 2

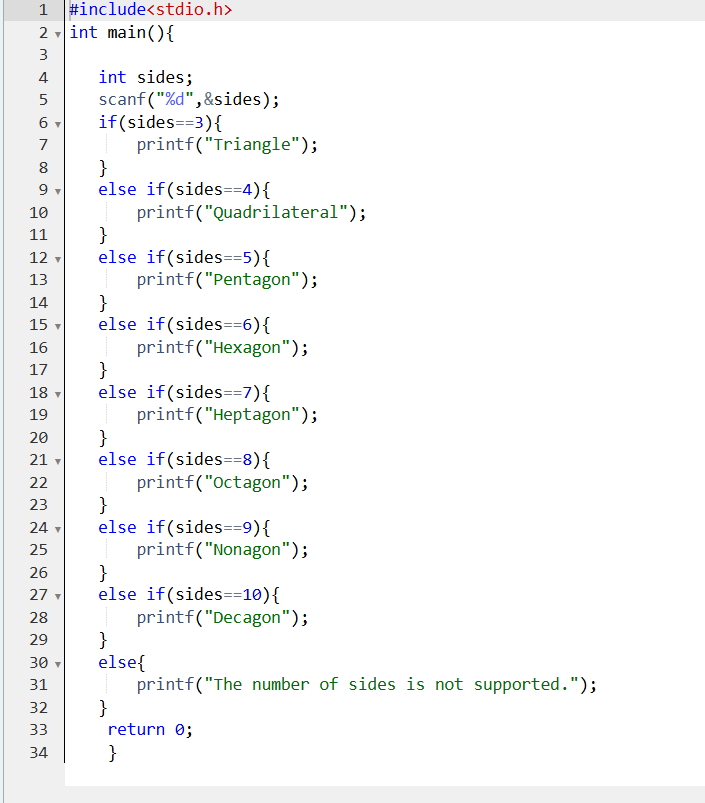
Heptagon

Sample Input 3

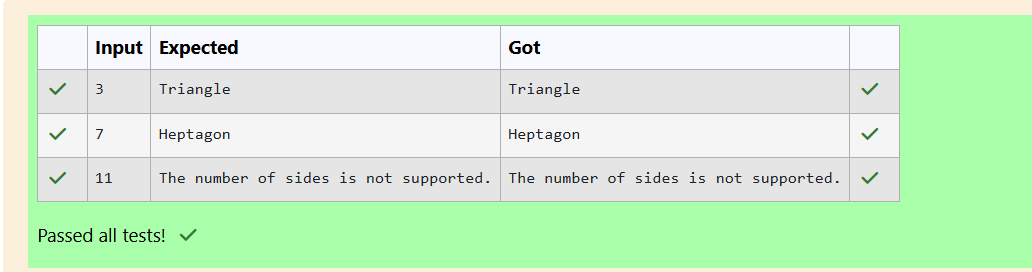
11

Sample Output 3

The number of sides is not supported



**OUTPUT:**



**2.** The Chinese zodiac assigns animals to years in a 12-year cycle. One 12-year cycle is shown in the table below. The pattem repeats from there with 2012 being another year of the Dragon, and 1999 being another year of the Hare

Year

Animal

2000

Dragon

2001

Snake

2002

Horse

2003

Sheep

2004

2005

Monkey

2006

Rooster

2007

Dog

2008

Pig

Rat

Ox

2009

2010

2011

Tiger

Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table

Sample Input 1

2004

Sample Output 1

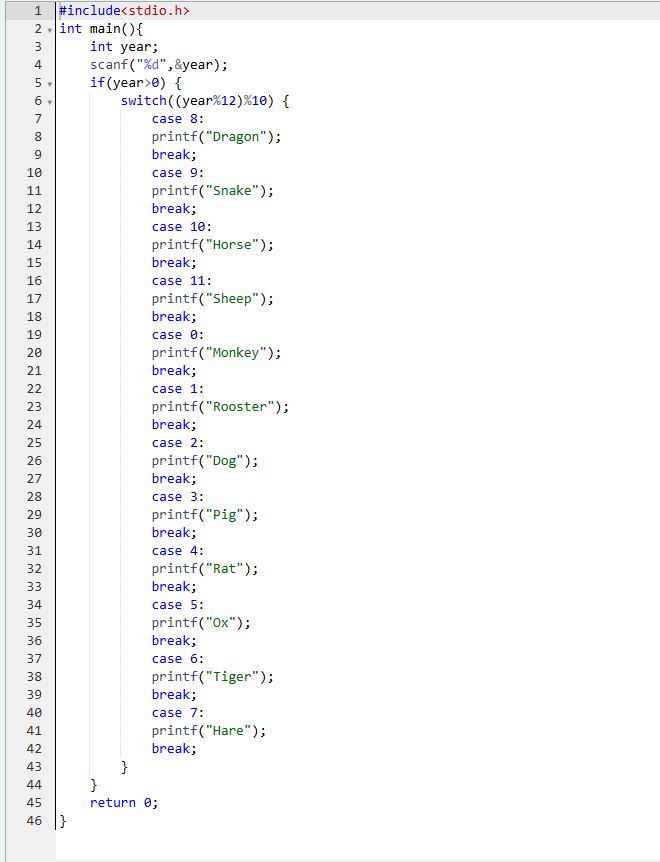
Monkey

Sample Input 2

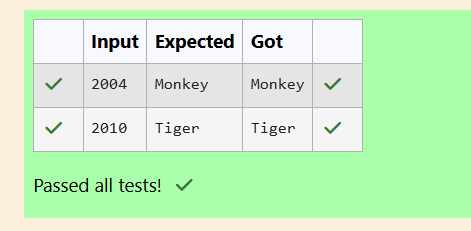
2010

Sample Output 2

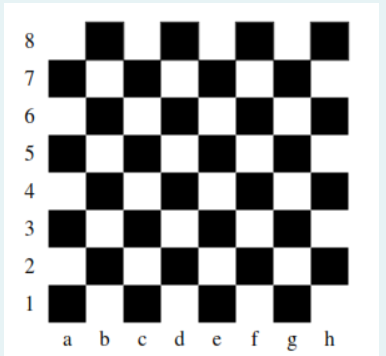
Tiger



**OUTPUT:**



1. Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:



Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square

Then use modular arithmetic to report the color of the square in that row. For example, if the user enters al then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

Sample Input 1

a1

[16-01-2025 22:38] Jeffery Antony: Sample Output 1

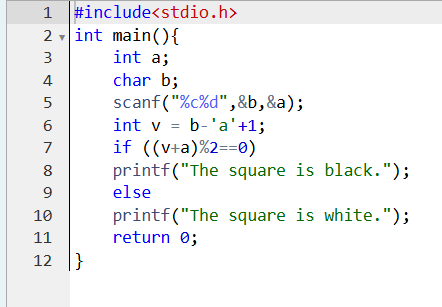
The square is black.

Sample Input 2

d5

Sample Output 2

The square is white.



**OUTPUT:**

