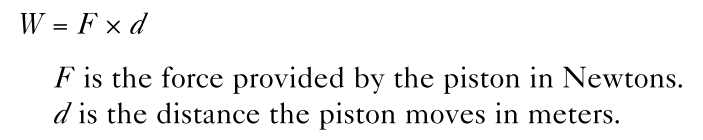
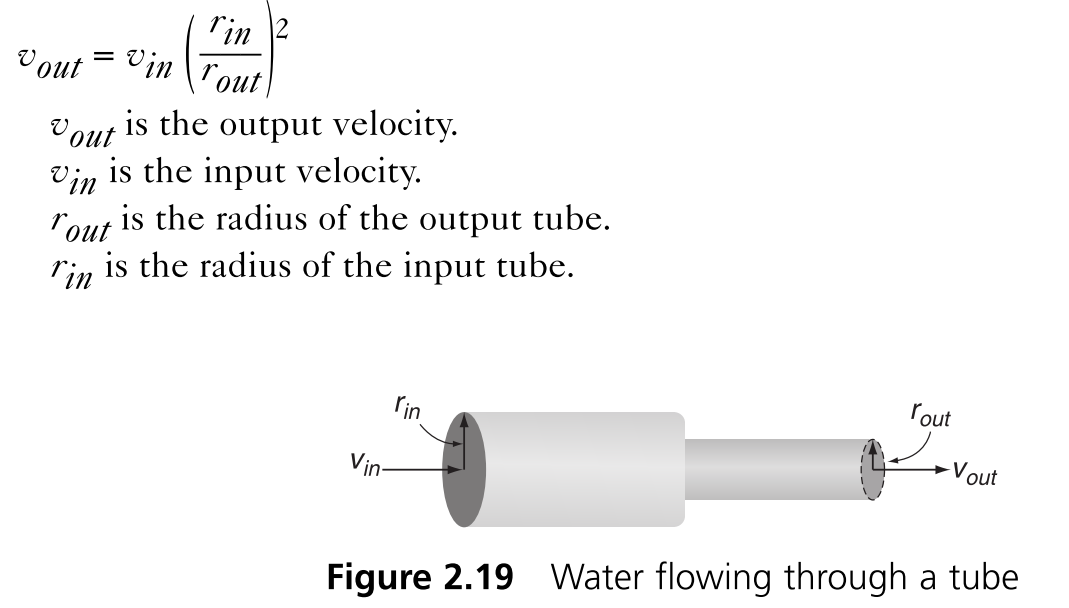
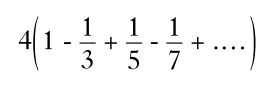
1. Design, write, compile, and run a program that determines the work, W, performed by a piston engine providing a force of 1000 N over a distance of 15 centimeters. The following formula is used to determine the work performed:



1. Design, write, compile, and run a C++ program that calculates and displays the velocity of water flowing out of the tube shown in Figure 2.19. The velocity of water flowing into the tube is 1 ft/sec, the input tube radius is 0.75 , and the output tube radius is 0.5 . The output velocity is given by this formula.



1. Write a C++ program to calculate the circumference of a circle having a radius of 3.3 inches. The formula for determining the circumference, c, of a circle is c = 2πr, where r is the radius and π equals 3.1416.
2. The value of π can be approximated by this series:



Using this formula, write a program that calculates and displays the value of π, using 2, 3, and 4 terms of the series.

1. Write, compile, and run a program to calculate and display the fourth root of a user-entered number. Recall from elementary algebra that you find the fourth root of a number by raising the number to the 1/4 power. (Hint: Don’t use integer division—can you see why?) Verify your program by calculating the fourth roots of this test data: 81, 16, 1, and 0. When you’re finished, use your program to determine the fourth roots of 42, 121, 256, 587, 1240, and 16,256。
2. Given an integer x, return true if x is palindrome integer.An integer is a palindrome when it reads the same backward as forward. For example, 121 is palindrome while 123 is not.

Example 1:

Input: x = 121

Output: true

Example 2:

Input: x = -121

Output: false

Explanation: From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a palindrome.

Example 3:

Input: x = 10

Output: false

Explanation: Reads 01 from right to left. Therefore it is not a palindrome.

Example 4:

Input: x = -101

Output: false

Constraints:-2^31 <= x <= 2^31 – 1

1. Find First and Last Position of Element in Sorted Array. Given an array of integers nums sorted in ascending order, find the starting and ending position of a given target value.If target is not found in the array, return [-1, -1].

Example 1:

Input: nums = [5,7,7,8,8,10], target = 8

Output: [3,4]

Example 2:

Input: nums = [5,7,7,8,8,10], target = 6

Output: [-1,-1]

Example 3:

Input: nums = [], target = 0

Output: [-1,-1]

Constraints:

0 <= nums.length <= 10^5

-10^9 <= nums[i] <= 10^9

nums is a non-decreasing array.

-10^9 <= target <= 10^9