Example Programs and Several Algorithms using Loops

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To learn loops from a variety of examples (GCD, FutureTuition, Dec2Hex, Monte Carlo Simulation) (§5.10 - §5.11).
- To write a program that displays prime numbers (§5.11).
- To write a program that checks whether a string is a palindrome (§5.13)

Outline

- Discussed
  - while Loop vs. do-while Loop vs. for Loop;
  - Nested loops
  - break and continue
  - Pitfalls and Errors
- Algorithms and Example Programs
  - Converting hexadecimal numbers to decimal numbers
  - Finding the Greatest Common Divisor
  - Finding the root of an equation
  - Simple Monte Carlo simulations
  - Palindromes
  - Prime numbers

Problem. Converting Decimals to Hexadecimals

- Hexadecimals are often used in computer systems programming (see Appendix for an introduction to number systems). Write a program to convert a hexadecimal number to the decimal number.

Solution. Converting Decimals to Hexadecimals

- To convert a decimal number *d* to a hexadecimal number is to find the hexadecimal digits h_n, h_{n-1}, h_{n-2}, ... , h_2, h_1, and h_0 such that

$$d = h_n \times 16^n + h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \ldots + h_2 \times 16^2 + h_1 \times 16^1 + h_0 \times 16^0$$

- These hexadecimal digits can be found by successively dividing *d* by 16 until the quotient is 0. The remainders are h_0, h_1, h_2, ... , h_{n-2}, h_{n-1}, and h_n.

Problem. Finding the Greatest Common Divisor

- Problem: Write a program that prompts the user to enter two positive integers and finds their greatest common divisor

Solution. Finding the Greatest Common Divisor

- Suppose you enter two integers 4 and 2, their greatest common divisor is 2. Suppose you enter two integers 16 and 24, their greatest common divisor is 8.
- So, how do you find the greatest common divisor?
- Let the two input integers be n1 and n2. You know number 1 is a common divisor, but it may not be the greatest commons divisor. So you can check whether k (for k = 2, 3, 4, and so on) is a common divisor for n1 and n2, until k is greater than n1 or n2

Problem. Predicting Future Tuition

- Problem: Suppose that the tuition for a university is $10,000 this year and tuition increases 7% every year. In how many years will the tuition be doubled?
- This is in fact a root finding problem, i.e., find x such that y = f(x).
- For this problem, we need to find n, such that, $20,000 = 10,000 * (1 + 0.07)^n$

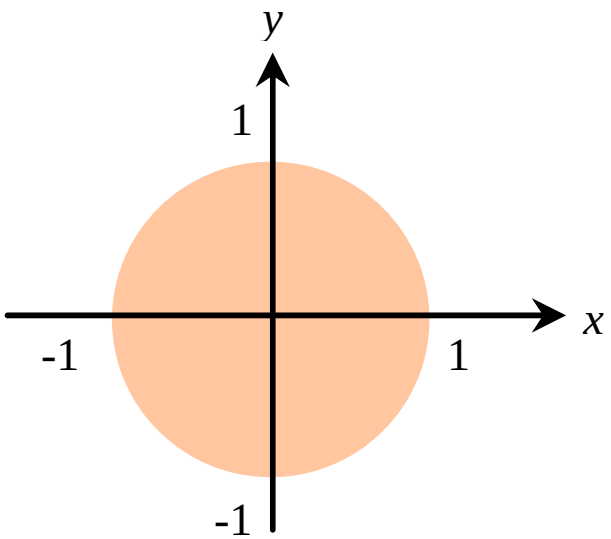
Solution. Predicting Future Tuition

- Compute tuition repeatedly for year 1, 2, …, until the tuition is greater than or equal to 20,000

Problem. Estimating π using Monte Caro Simulation

- The Monte Carlo simulation refers to a technique that uses random numbers and probability to solve problems.
- This method has a wide range of applications in computational mathematics, physics, chemistry, and finance.
- Let's consider to use the Monto Carlo simulation for estimating π

Solution. Estimating π using Monte Caro Simulation



circleArea / squareArea = π / 4.

π can be approximated as 4 * numberOfHits / numberOfTrials

Questions

- Can you design solutions for these problems using
  - While loops
  - Do-while loops
  - For any loops
  - Combinations of the above loops

Guessing Number Problem Revisited

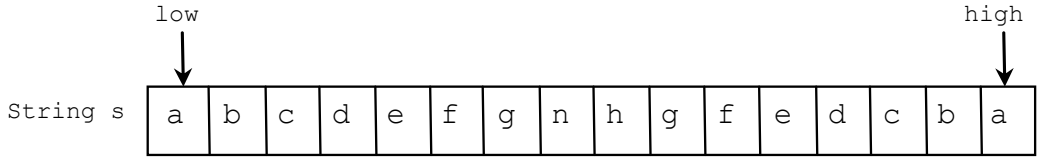
- “break” out of the loop when the guess is correct.

Problem. Checking Palindromes

- A string is a palindrome if it reads the same forward and backward. The words “mom,” “dad,” and “noon,” for instance, are all palindromes.
- The problem is to write a program that prompts the user to enter a string and reports whether the string is a palindrome.

Solution. Checking Palindromes

- Check whether the first character in the string is the same as the last character. If so, check whether the second character is the same as the second-to-last character.
- This process continues until a mismatch is found or all the characters in the string are checked, except for the middle character if the string has an odd number of characters.



Problem. Displaying Prime Numbers

- Problem: Write a program that displays the first 50 prime numbers in five lines, each of which contains 10 numbers. An integer greater than 1 is *prime* if its only positive divisor is 1 or itself. For example, 2, 3, 5, and 7 are prime numbers, but 4, 6, 8, and 9 are not.

Solution. Displaying Prime Numbers

- Solution: The problem can be broken into the following tasks:
  1. For number = 2, 3, 4, 5, 6, ..., test whether the number is prime.
  2. Determine whether a given number is prime.
  3. Count the prime numbers.
  4. Print each prime number, and print 10 numbers per line.

Questions