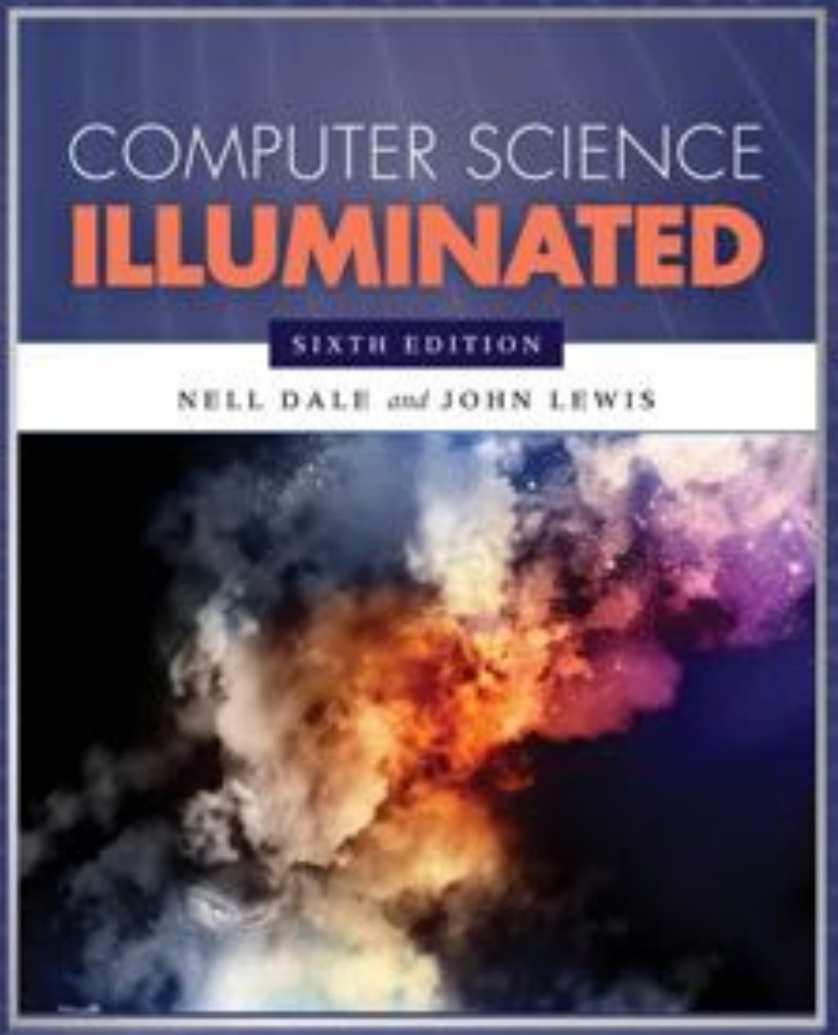
Pseudocode



Chapter Goals

- Distinguish between following an algorithm and developing one
- Describe the pseudocode constructs used in expressing an algorithm
- Use pseudocode to express an algorithm
- Describe two approaches to testing
- Design and implement a test plan for a simple assembly-language program

Pseudocode

Pseudocode

A mixture of English and formatting to make the steps in an algorithm explicit

Algorithm to Convert base-10 number to other bases

While (the quotient is not zero)
Divide the decimal number by the new base
Make the remainder the next digit to the left in the answer
Replace the original decimal number with the quotient

Following an Algorithm

Never-Fail Blender Hollandaise

1 cup butter
4 egg yolks
1/4 teaspoon salt
1/4 teaspoon sugar
1/4 teaspoon Tabasco
1/4 teaspoon dry mustard
2 tablespoons lemon juice

Heat butter until bubbling. Combine all other ingredients in blender. With blender turned on, pour butter into yolk mixture in slow stream until all is added. Turn blender off. Keeps well in refrigerator for several days. When reheating, heat over hot—not boiling—water in double boiler. Makes about 1-1/4 cups sauce.

Following an Algorithm

Algorithm for preparing a Hollandaise sauce

IF concerned about cholesterol
    Put butter substitute in a pot
ELSE
    Put butter in a pot
Turn on burner
Put pot on the burner
WHILE (NOT bubbling)
    Leave pot on the burner
Put other ingredients in the blender
Turn on blender
WHILE (more in pot)
    Pour contents into lender in slow steam
Turn off blender

Developing an Algorithm

Two methodologies used to develop computer solutions to a problem

- Top-down design focuses on the **tasks** to be done
- Object-oriented design focuses on the **data** involved in the solution

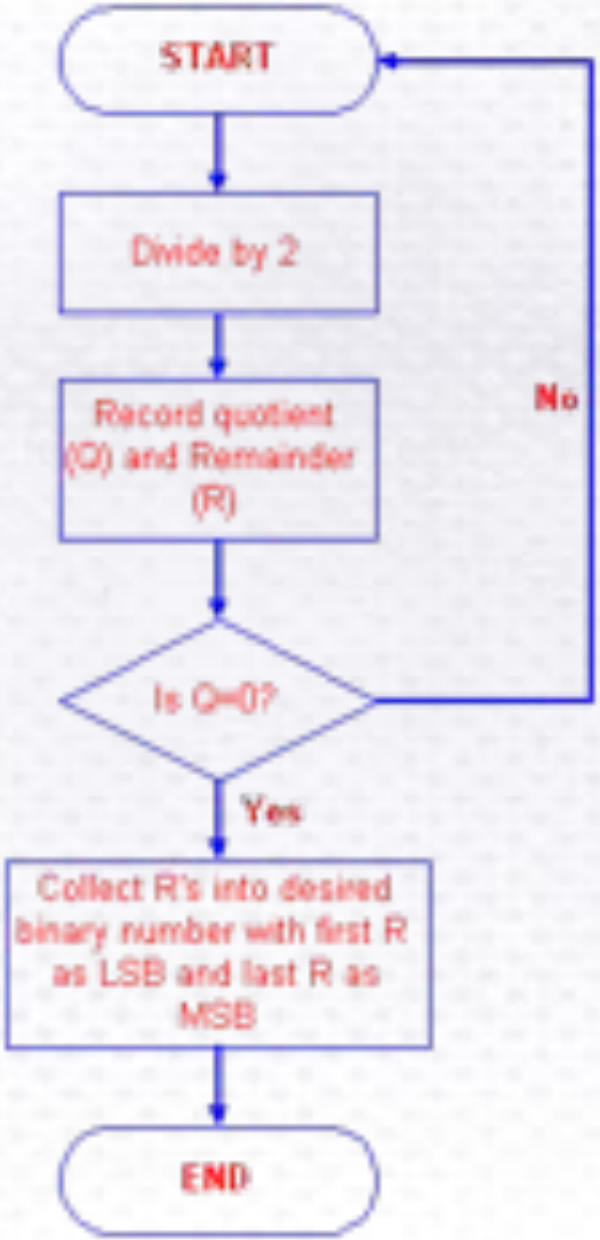
But first, let's look at a way to express algorithms: pseudocode

Pseudocode

Pseudocode

A way of expressing algorithms that uses a mixture of *English phrases* and *indentation* to make the steps in the solution explicit

There are no grammar rules in pseudocode, but it's important to be consistent and unambiguous



Following Pseudocode

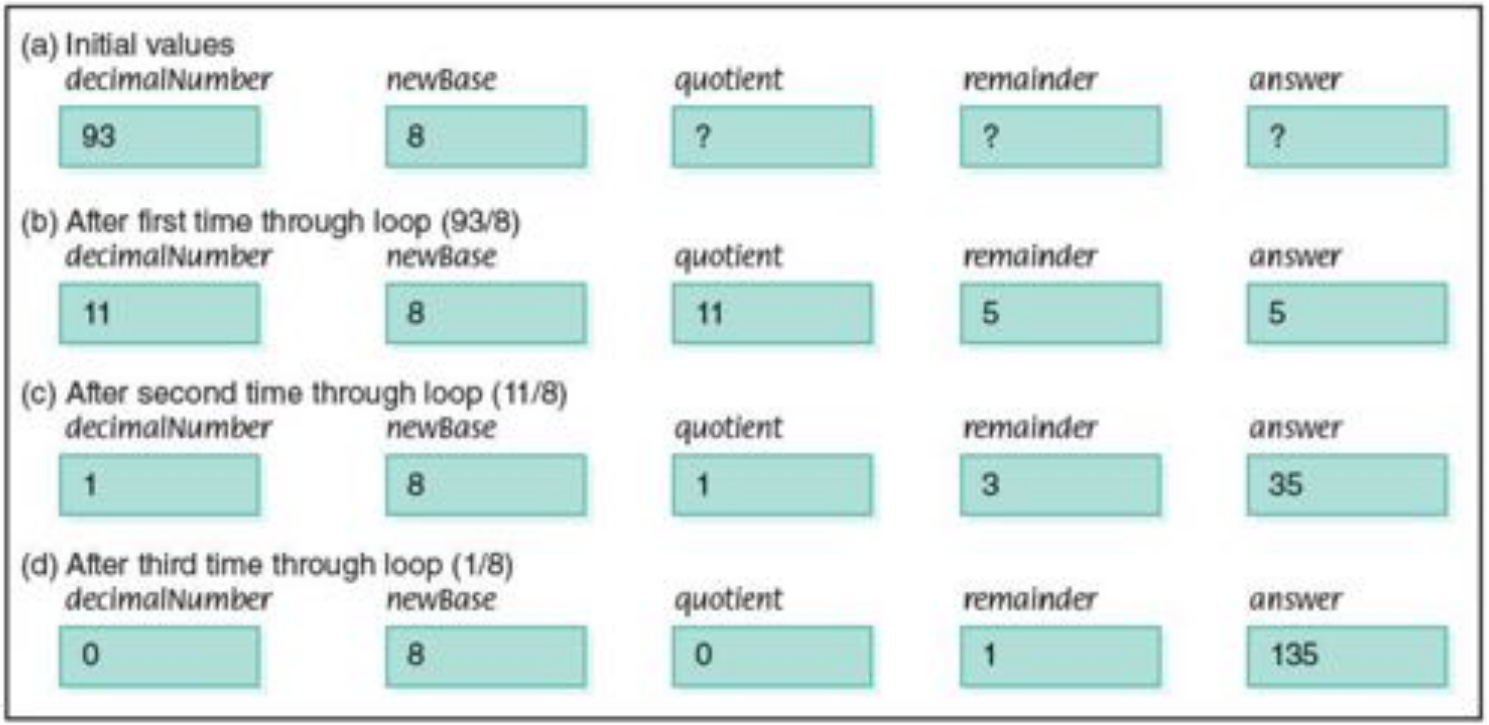
While (the quotient is not zero)
Divide the decimal number by the new base
Make the remainder the next digit to the left in the answer
Replace the original decimal number with

What is 93 in base 8?

93/8 gives 11 remainder 5
11/8 gives 1 remainder 3
1/ 8 gives 0 remainder 1

answer 1 3 5

Following Pseudocode



(a) Initial values

decimalNumber	*newBase*	*quotient*	*remainder*	*answer*
93	8	?	?	?

(b) After first time through loop (93/8)

decimalNumber	*newBase*	*quotient*	*remainder*	*answer*
11	8	11	5	5

(c) After second time through loop (11/8)

decimalNumber	*newBase*	*quotient*	*remainder*	*answer*
1	8	1	3	35

(d) After third time through loop (1/8)

decimalNumber	*newBase*	*quotient*	*remainder*	*answer*
0	8	0	1	135

FIGURE 6.6 Walk-through of a conversion algorithm

Easier way to organize solution

Pseudocode for Complete Computer Solution

Write "Enter the new base"

Read newBase

Write "Enter the number to be converted"

Read decimalNumber

Set quotient to 1

WHILE (quotient is not zero)

    Set quotient to decimalNumber DIV newBase

    Set remainder to decimalNumber REM newBase

    Make the remainder the next digit to the left in the answer

    Set decimalNumber to quotient

Write "The answer is "

Write answer

Pseudocode Functionality

Variables

Names of places to store values

quotient, decimalNumber, newBase

Assignment

Storing the value of an expression into a variable

Set quotient to 64

quotient <-- 64

*quotient <-- 6 * 10 + 4*

Pseudocode Functionality

Output

Printing a value on an output device

Write, Print

Input

Getting values from the outside word and storing them into variables

Get, Read

Pseudocode Functionality

Selection

Making a choice to execute or skip a statement (or group of statements)

```
Read number
IF (number < 0)
        Write number + " is less than zero."
```

or

```
Write "Enter a positive number."
Read number
IF(number < 0)
        Write number + " is less than zero."
        Write "You didn't follow instructions."
```

Pseudocode Functionality

Selection

Choose to execute one statement (or group of statements) or another statement (or group of statements)

```
IF ( age < 12 )
        Write "Pay children's rate"
        Write "You get a free box of popcorn"
ELSE IF ( age < 65 )
        Write "Pay regular rate"
ELSE
        Write "Pay senior citizens rate"
```

Pseudocode Functionality

Repetition

Repeating a series of statements

```
Set count to 1
WHILE ( count < 10)
        Write "Enter an integer number"
        Read aNumber
        Write "You entered " + aNumber
        Set count to count + 1
```

How many values were read?

Pseudocode Example

Problem: Read in pairs of positive numbers and print each pair in order.

WHILE (not done)
Write "Enter two values separated by blanks"
Read number1
Read number2
Print them in order

Pseudocode Example

How do we know when to stop?

Let the user tell us how many

Print them in order?

If first number is smaller

print first, then second

Otherwise

print second, then first

Pseudocode Example

```
Write "How many pairs of values are to be entered?"
Read numberOfPairs
Set numberRead to 0
WHILE (numberRead < numberOfPairs)
        Write "Enter two values separated by a blank; press return"
        Read number1
        Read number2
        IF(number1 < number2)
                Print number1 + " " + number2
        ELSE
                Print number2 + " " + number1
        Increment numberRead
```

Walk Through

Data

3

55 70

2 1

33 33

numberOfPairs

Fill in values during each iteration

numberRead	*number1*	*number2*

What is the output?

Translating Pseudocode

To What?

- Assembly language
  - Very detailed and time consuming
- High-level language
  - Easy, as you'll see in Chapter 9

Testing

Test plan

A document that specifies how many times and with what data the program must be run in order to thoroughly test it

Code coverage

An approach that designs test cases by looking at the code

Data coverage

An approach that designs test cases by looking at the allowable data values

What does "thoroughly" mean?

Testing

Test plan implementation

Using the test cases outlined in the test plan to verify that the program outputs the predicted results

Important Threads

Operations of a Computer

Computer can store, retrieve, and process data

Computer's Machine Language

A set of instructions the machine's hardware is built to recognize and execute

Machine-language Programs

Written by entering a series of these instructions in binary form

Important Threads

Pep/8: A Virtual Computer with One Register (A) and two-part instructions:

One part tells which action the instruction performs; the other part details where the data to be used can be found

Pep/8 Assembly Language

A language that permits the user to enter mnemonic codes for each instruction rather than binary numbers

Pseudocode

Shorthand-type language people use to express algorithms

Important Threads

Testing Programs

All programs must be tested; code coverage testing and data coverage (black-box testing) are two common approaches

Ethical Issues

Software Piracy and Copyrighting

Have you ever "borrowed" software from a friend?

Have you ever "lent" software to a friend?

According to IDC, lowering software piracy by 10% over the next four years would create 500,000 jobs

Who am I?

© Karl Staedele/dpa/Corbis

Turing, Atanasoff, Eckert, and Mauchly were my contemporaries. Why were we unaware of each other's work?

Do you know?

How is a computer data base helping endangered species?

What would chess grandmaster Jan Helm Donner do if he had a hammer?

What are Nigerian check scams?

Why does an anthropologist work for Intel?

What is the Music Genome Project?

What music streaming website uses the Music Genome Project?

What is the difference between certification and licensing?