
Algorithm Discovery and Design

Topics:

Algorithmic Problem Solving
Conditional Operations

8.11.2007

Objectives

By the end of this lecture, you should be able to:

- Design algorithms using conditional operations

Algorithms: Operations

Algorithms can be constructed by the following operations:

- **Sequential Operation**
- **Conditional Operation**
- **Iterative Operation**

Conditional Operation

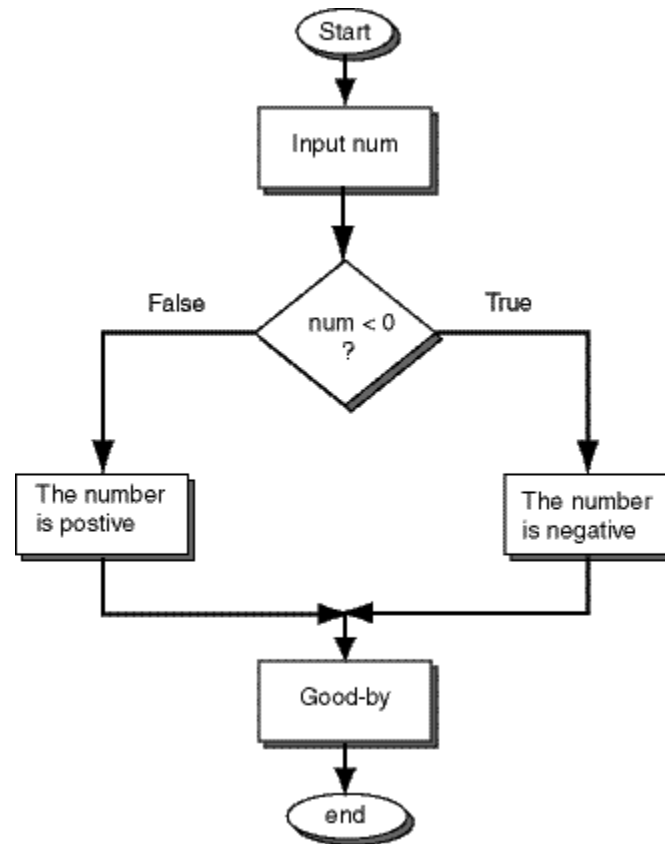
- Determines whether or not a condition is true; and based on whether or not it is true; **selects the next step** to do
- Use the same primitives as before plus the following:

```
if <condition> then
    operations for the then-part
else
    operations for the else-part
endif
```

Execution

- Evaluate <condition> expression to see whether it is true or false.
- If true, then execute operations in then-part
- Otherwise, execute operations in else-part

Conditional Operation



```
get num
if ( num < 0 )
    print "The number is negative"
else
    print "The number is positive"
endif
print "Good-bye for now"
```

Conditional Operation

Example 1: Give the user a choice of seeing the area or the circumference of a circle given its radius.

```
get radius
print "Type A for area or C for circumference"
get response
if (response = A) then
    set area to (radius * radius * 3.14)
    print area
else
    set circumference to (2 * radius * 3.14)
    print circumference
endif
```

Conditional Operation

- **Example 2:** Write an algorithm to convert EURO to Egyptian Pound (L.E.) and Egyptian Pound to EURO. The inputs to your algorithm are the following:
 - Amount of money to be converted
 - Conversion Type (1 for EURO to L.E. and 2 for L.E. to EURO)
 - Exchange Rate (i.e.: the L.E. equivalent for 1 EURO)

```
get amount, type, rate
if (type=1) then
    set amount to amount * rate
else
    set amount to amount / rate
endif
print amount
```

Compounded Conditions

- Conditions may be **compounded** using AND, OR and NOT
 - E1 or E2: true if at least one of them is true; false otherwise.
 - E1 and E2: true if both are true; false otherwise.
 - not E: true if E is false, false, if E is true.
 - **Example:** Find the sum of three positive numbers

```
get A, B, C
if (A>0) and (B>0) and (C>0) then
    set Sum to (A+B+C)
    print Sum
endif
```


Conditional Algorithms with more than two choices

Nested If statement:

```
if first condition then
    // do first thing
else    if second condition then
        // do second thing
    else
        // do something else
    endif
endif
```

Nested If Statement

- **Example 1:** Algorithm to find the largest of three numbers.

```
get A, B, C
if (A>=B) then
    if (A>=C) then
        print A
    else
        print C
    endif
else
    if (B>=C) then
        print B
    else
        print C
    endif
endif
```

Nested If Statement

- **Example 2:** The owner of a bookshop has his own rules for deciding how many copies, if any, he will order of each book. Using only the information given below, specify an algorithm using pseudo-code to find the order required for one book.
 - Reprinted books: don't order any copies.
 - Non-fiction: order 50 copies.
 - Fiction: order 300 copies unless the book is written by a best selling author, in which case order 900 copies.

Nested If Statement

```
get type, author
if (type = reprinted)
    print 0
else
    if (type = nonfiction)
        print 50
    else
        if (type = fiction)
            if (author = best selling)
                print 900
            else
                print 300
            endif
        else
            print "bad data"
        endif
    endif
endif
endif
```