

Stepwise Decomposition and Flowcharts

This lecture introduces two techniques that are useful in developing and documenting algorithms: stepwise decomposition and flowcharts.

After this lecture you should be able to: (1) reduce a familiar but complex procedure into a sequence of simpler steps, (2) describe an algorithm using a flowchart, and (3) convert back and forth between a flowchart and C code.

Stepwise Decomposition

An algorithm is a sequence of steps used to solve a particular problem. Developing algorithms is an important part of writing programs.

An effective way to develop an algorithm is to reduce the initial problem into a sequence of simpler problems. These problems can themselves be broken down into simpler problems. The process can be repeated as required.

For example, changing a tire can be broken down into two steps: (1) remove old tire, (2) put on new tire. Each of these steps can be further decomposed into simpler steps, for example the 'remove old tire' step could be broken down into: remove hubcap, loosen bolts, raise car, remove nuts, remove tire. Each of these steps can then be decomposed into smaller steps. For example, 'remove nuts' can be broken down into: for each nut: put wrench on nut, turn wrench counterclockwise until nut comes off .

This process of decomposition can be carried down to any desired level of detail.

Exercise: Break down the task of buying a candy bar from a vending machine into a sequence of three steps. Break each of these steps into three simpler steps.

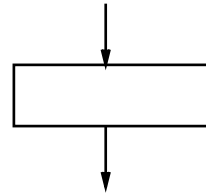
When developing an algorithm for a computer program the decomposition procedure is repeated until the remaining actions are simple enough to express as statements in the programming language.

Therefore in order to express an algorithm in 'C' it is necessary to reduce the problem until each step is one of the following: (1) evaluating an expression, (2) repeating other step(s) while a condition is true (a `while` statement), and (3) executing one step or another depending on a condition (a `if/else` statement).

Flowcharts

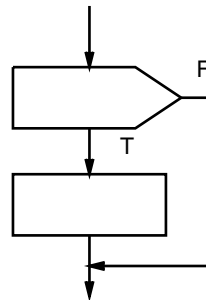
A flowchart is a graphical representation of an algorithm. Symbols of different shapes are used for the three different basic types of statements and lines join the symbols to indicate the sequence in which the statements are executed (the "flow of control").

The symbol used for an expression statement is a rectangular block:



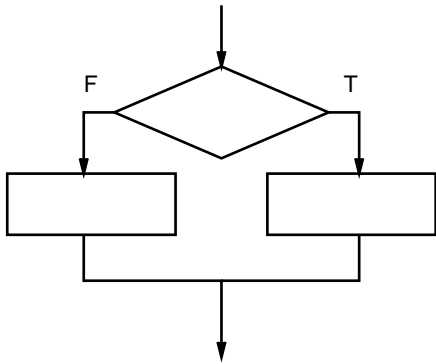
with the expression written inside the block.

The symbol for a `while` statement is:



with an expression written inside the top block. The block below the `while` block is executed while the expression is non-zero (T) and not executed when the expression is zero (F).

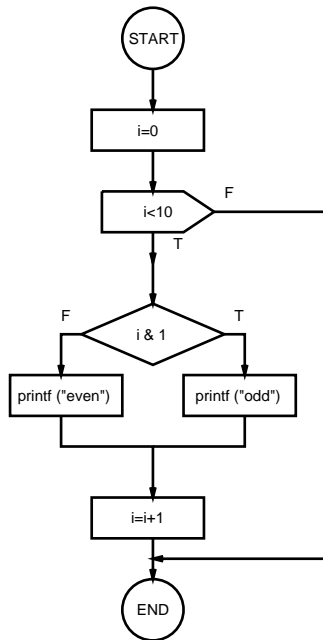
The symbol for a `if/else` statement is:



with an expression written inside the diamond symbol. The right hand block is executed if the expression is non-zero (T) and the left block is executed if the expression is zero (F).

Note that each of the three above symbols has one line going into it from the top and one line leaving it from the bottom. The process of stepwise decomposition involves joining sequences of these symbols and iteratively replacing each block with a `while` or `if/else` symbols.

Exercise: Write the C code corresponding to the following flowchart:



Exercise: Convert the following C code to a flowchart:

```

if ( i > 0 ) {
    while ( i < 10 ) {
        printf ( "%d", i ) ;
        i = i + 1 ;
    }
} else {
    i = i + 1 ;
}

```