

Writing Algorithms

This handout briefly describes a 4 step approach to constructing algorithms. The steps are outlined below.

1) Understand the problem or what the question ask

A common mistake is to begin writing an algorithm before the problem to be solved is fully understood. Among other things, understanding the problem involves knowing the answers to the following questions.

What is the input list, i.e. what information will be required by the algorithm?

What is the desired output or result?

2) Devise a Plan

Remember: you cannot write instructions for solving a task if you can't perform the task yourself. Thus, the first step in devising a plan is to perform the task yourself. Arm yourself with paper and pencil and try to determine precisely what steps are used when you perform the task in your own mind. Force yourself to perform the task slowly and methodically – think about *what you are doing*. Do not use your own memory – when you need to remember a quantity, introduce variables (named quantities) that can be used to hold information.

Using the information gleaned from solving the problem by hand, devise an explicit step-by-step method of solving the problem. These steps can be written down in an informal outline form – it is not necessary at this point to use pseudo-code.

3) Refine the Plan and Translate into Pseudocode

The plan devised in step 2) needs to be translated into pseudocode. This refinement may be trivial or it may take several “passes”.

4) Test the Design

Using appropriate test data, test the algorithm by going through it step by step. Think about ordinary sets of input data as well as those which are “degenerate”. If you find errors correct them. Some errors may be so serious that you should scrap your pseudocode and go back to step 2).

A common and **big mistake** is to skip this step and go blindly forward in the programming process, eventually translating an incorrect algorithm into an incorrect program.