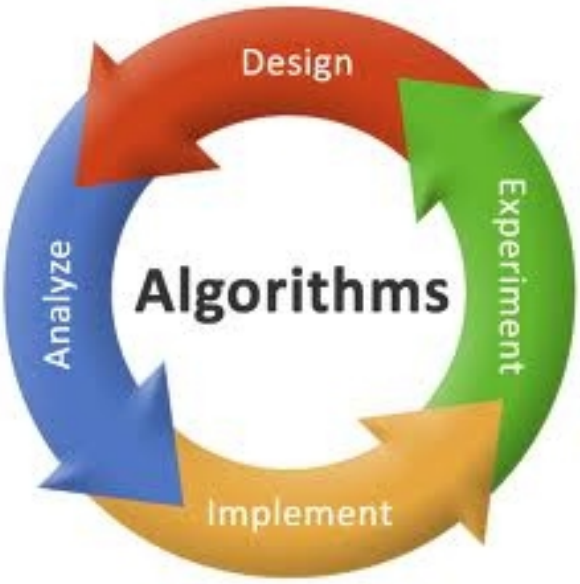
Chapter 2



ALGORITHMs

You'll learn about the algorithm in this chapter, which is necessary in the design of solutions to problems and which programmers can code in any number of languages.

After completing this unit you will be able to know:

- algorithm and its relationship to programming;
- the three parts of an algorithm and
- make algorithms to real world problems.

F. THE ALGORITHM: The Basis for All Designs to Solutions of Programming Problems

We are almost ready to begin the subject of programming. Taking problems in the real world and writing them in a language that the computer can "understand" or execute. Before you get to the stage of programming your problems, you must design a suitable way of solving problems. An algorithm is a set of steps for solving a problem. These steps may repeat and may involve some decisions, such as a choice of two or more things.

Consider the following example of an algorithm for buying a ticket to a movie.

1. Go to the theater.
2. Walk to the ticket counter.
3. Select a movie.
4. Pay the price.
5. Receive the ticket.

What about an algorithm for finding the smallest number among three numbers?

1. Compare the first number with the second number.
2. Discard the bigger number from step 1.
3. Compare the third number with the number that's left.
4. Discard the bigger number from step 3.
5. Whatever number is left is the smallest of all three.

The Three Parts of Any Algorithm

An algorithm has three parts:

1. The steps are finite (i.e., they do not go on forever).
2. Steps may be repeated.
3. Steps may involve decision making.

Each step of an algorithm should follow the step before. If necessary, repeat some of the steps and skip others if a decision calls for that action.

Examples of Algorithms in the Real World

Some examples of other algorithms follow.

Example1 : Buying Fries at McDonalds

1. Select the size of fries that you want.
2. Take out enough money to pay for them.
3. Hand over the money.
4. Receive change if necessary.
5. Take fries with you.

Example 2: Dialing a Friend Who Lives Locally

1. Pick up the receiver.
2. Dial the six digits number.
3. Wait for your friend to answer the call.

EXAMPLE 3. Let's find an algorithm to calculate the year-end bonus for all salaried Employees are to be paid 3% of their annual salary or 4,000, whichever is larger.

To carry out this task, a payroll clerk might proceed as follows.

a. Open the employee ledger.
b. Turn to the next employee's account.
c. Determine the employee's bonus.
d. Write the employee's name and bonus amount on the bonus sheet.
e. If all bonuses have not been determined, return to step (b).
f. Close the ledger

It is not difficult to see that these six instructions constitute an algorithm. Each step is well defined, and, since a business can employ only a finite number of people, the algorithm will terminate in a finite number of steps. Moreover, if this algorithm is followed, all employee bonuses will be determined as specified.

Although the algorithm "does" what was asked, the process could be made more specific by including more detail in step (c). Recalling the method specified for calculating bonus amounts, we can substitute the following for step (c).

c1. Multiply the employee's salary by .03 to obtain a tentative bonus.
c2. If the tentative bonus is at least 4,000, go to step (d).
c3. Set the bonus to 4,000.

Making this change, or refinement, we obtain the following more detailed algorithm.

a. Open the employee ledger.
b. Turn to the next employee's account.

c1. Multiply the employee's salary by .03 to obtain a tentative bonus.
c2. If the tentative bonus is at least 4,000, go to step (d).
c3. Set the bonus to 4,000.

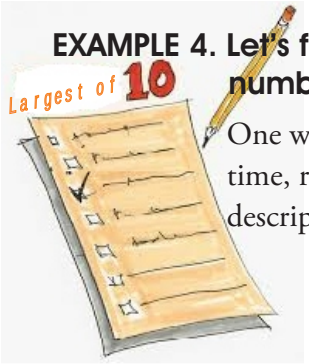
d. Write the employee's name and bonus amount on the bonus sheet.
e. If all bonuses have not been determined, return to step (b).
f. Close the ledger.

Remark 1

The first algorithm is more general than the second. It describes a process one might follow to determine employee bonuses, however they are to be calculated. The second algorithm can be used only if bonuses are calculated as specified in the problem statement.

Remark 2

It is somewhat easier to verify that the first algorithm "does" what was asked. This is because of, and not in spite of, the detail not present in the algorithm. Having verified that the first algorithm is correct, all that is required to verify that the second algorithm is also correct is to check that steps (c1), (c2), and (c3) describe the same task as step (c) of the first algorithm.



EXAMPLE 4. Let's find an algorithm to determine the largest number in a list of ten numbers.

One way to determine the largest number in a list of ten numbers is to read them one at a time, remembering only the largest of those already read. To help us give a precise description of this process, let's use two symbols as follows:

L to denote the largest of those numbers already read.
N to denote the number currently being read.

The following algorithm can now be written.

a. Read the first number and denote it by L.
b. Read the next number and denote it by N.
c. If L is at least as large as N, skip the next step.
d. Assign the numbers have not been read, go to step (b).
e. If all ten numbers have not been read, go to step (b).
f. Print the value of L and stop.

To verify this algorithm for the list of numbers.

4,5,3,6,6,2,1,8,7,3

We simply proceed step-by-step through the algorithm, always keeping track of the latest values of L and N. An orderly way to do this is to complete an assignment table as follows.

Algorithm Step	L	N
a	4	
b		5
d	5	
b		3
b		6
d	6	
b		6
b		2
b		1
b		8
d	8	
b		7
b		3

Note that after all ten numbers have been processed the largest number in the list is the last value of L.

Remark If the ten numbers are written on a sheet of paper, a person could simply look them over and select the largest. However, this process is heuristic and does not constitute an algorithm. To see that this is so, imagine many hundreds of numbers written on a large sheet of paper. In this case, selecting the largest simply by looking over the numbers can easily result in an error. What is needed is an orderly process that will ensure that the largest number is selected. Examining numbers one at a time, as was done in the algorithm, is such an orderly process.

Snap Quiz

a written test

1. Problem Solving

Problems 1-4 refer to the following algorithm, which is intended for completing an invoice:

a. Let AMOUNT = 0.
b. Read QUANTITY and PRICE of an item.
c. Add the product QUANTITY x PRICE to AMOUNT.
d. If there is another item, go to step (b). Otherwise, continue with step (e).
e. If AMOUNT is not greater than 500, go to step (h). Otherwise, continue with step (f).
f. Evaluate the product .05 x AMOUNT, g. Subtract this product from AMOUNT.
h. Record the value AMOUNT and stop.

1. What interpretation could be given to the product appearing in step (f)?

2. What purpose would you say is served by step (e)?

3. If the values (10, 3.00), (50, 8.00), and (25, 12.00) are read by step (b), what value will be recorded by step (h)?

4. If the values (100, 2.00) and (50, 1.00) are read by step (b), what value will be recorded by step (h)?

Problems 5-8 refer to the following algorithm, which is intended for use by a payroll clerk as the preliminary step in the preparation of a payroll:

a. Read the next time card.
b. Let H = number of hours worked.
c. If H is not greater than 32, let G = B = 0 and go to step (f). Otherwise, continue with the next step.
d. Evaluate 6 x (H - 32) and assign this value to both G and B.
e. Let H = 32.
f. Evaluate 4 x H and add this value to G.
g. Write the values G and B on the time card,
h. If there is another time card, go to step (a). Otherwise, stop.

5. If the numbers of hours shown on the first four time cards are 20, 32,40, and 45, respectively, what amounts will be written on these cards?

6. What is the base hourly rate for each employee?

7. What is the overtime rate?

8. Explain step (c).

TEAR THIS PAGE AND SUBMIT IT TO YOUR TEACHER FOR RECORDING.

What will be printed when each algorithm shown in **Problems 9-12** is carried out?

9. a. Let SUM = O and N = 1.
   b. Add N to SUM.
   c. If N<6, increase N by 1 and return to step (b). Otherwise, continue with step (d).
   d. Print the value SUM and stop.

10. a. Let PROD = N = 1.
    b. Print the values N and PROD on one line.
    c. Increase N by 1.
    d. If N exceeds 6, stop. Otherwise, continue with the next step.
    e. Multiply PROD by N and go to step (b).

11. a. Let A = B = 1 and NUM = 3.
    b. Evaluate A + B and assign this value to F.
    c. If NUM does not exceed 9, increase NUM by 1 and proceed to step (d). Otherwise, print the value F and stop, d. Assign the values of B and F to A and B, respectively, and go to step (b).

12. a. Let NUM = 56, SUM = T, and D = 2.
    b. If D is a factor of NUM, add D to SUM and print D.
    c. lf D $\geq$ NUM/2, print SUM and stop.
    d. Increase D by 1, and go to step (b).

Write an algorithm to carry out each task specified in **Problems 13-19**.

13. A retail store's monthly sales report shows, for each item, the fixed cost, the sale price, and the number sold. Prepare a three-column report with the column headings ITEM, GROSS SALES, and INCOME.

14. Each of several three-by-five cards contains an employee's name. Social Security number, job classification, and date hired. Prepare a report showing the names, job classifications, and complete years of service for employees who have been with the company for more than ten years.

15. A summary sheet of an investor's stock portfolio shows, for each stock, the corporation name, the number of shares owned, the current price, and the earnings as reported for the most recent year. Prepare a six-column report with the column headings CORP. NAME. NO. OF SHARPS. PRICE, EARNINGS, EQUITY, and PRICE/EARNINGS. Use this formula:

    Equity = No. of Shares x Price

16. Each of several cards contains a single number. Determine the sum and the average of all of the numbers. (Use a variable N to count how many cards are read and a variable SUM to keep track of the sum of the numbers already read.)

17. Each of several cards contains a single number. On each card, write the letter G if the number is greater than the average of all of the numbers. Otherwise, write the letter L on the card. (You must read through the cards twice: once to find the average, and a second time to determine whether to write the letter G or the letter L on the cards.)

18. A local supermarket has installed a check-validation machine. To use this service, a customer must have previously obtained an identification card containing a magnetic strip and also a four-digit code. Instructions showing how to insert the identification card into a special magnetic-strip reader appear on the front panel. To validate a check, a customer must present the identification card to the machine, enter the four-digit code, enter the amount of the check, and place the check, blank side toward the customer, in a special clearly labeled punch unit. To begin this process, the CLEAR key must be depressed, and, after each of the two entries has been made, the ENTER key must be depressed. Prepare an algorithm giving instructions for validating a check.

19. Write an algorithm describing the steps to be taken to cast a ballot in a national election. Assume that a person using this algorithm is a registered voter and has just entered the building in which voting is to take place. While in the voting booth, the voter should simply be instructed to vote. No instructions concerning the actual filling out of a ballot are to be given.

2. True or False. Shade A if the statement is correct and B if not, in the answer sheet provided.

1. The terms algorithm and process are synonymous.

2. The following steps describe an algorithm, a. Let N = 0. b. Increase N by 10. c. Divide N by 2. d. If N< 10, go to step (b). e. Stop.

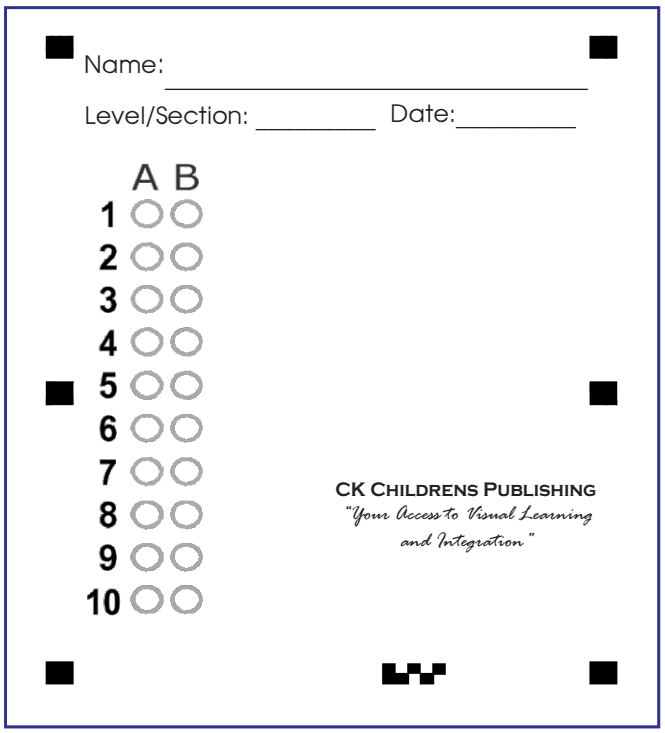
3. A computer program should describe an algorithm.

4. Every algorithm can be translated into a computer program.

5. The expression heuristic process refers to an algorithm.

6. It is always easier to verify the correctness of an algorithm that describes a very specific task than the correctness of a more general algorithm.

7. The expressions variable quantity and variable are used synonymously. Both refer to a quantity that can be changed during a process.



Programming

Now we are ready to discuss the subject of programming—taking a problem or task and designing an algorithm to handle this task, then using a programming language to express that algorithm so the computer will be able to execute that code.

Most people think of programming as being just about code. The lines of symbols and words that all of us have seen if we have ever opened a book on programming. It is more than code, however; it is a way of thinking about a problem and designing a solution that can then be written in a programming language. Look at this short C++ program that prints the message "Could you please say that again?" 250 times on the machine.

Sample of a C++ Code

```
# include <iostream.h>
# include <string.h>
int main ()
{ int x;
string first_phrase;
first_phrase = "Could you please say that again?";
for (x = 0; x < 250; x++)
 {
  cout << first_phrase << endl;
 }
return 0;
}
```

Sample code from Scratch

Here is another version of the same program in Pascal.

Sample of a Turbo Pascal Code

```
program printmessage ()
var
x : integer; first_phrase :string;
begin
first_phrase := "Could you please say that again?";
for x := 0 to 250 do
 begin
  writeln (first_phrase);
  x := x + 1;
 end
end
```

The point of showing both of these programs is to focus on how they are similar rather than their language differences. As you move through the chapters that follow, keep in mind that languages will always change to suit the development of technology. Certain concepts remain the same regardless of language. If you learn these concepts well, you will have no problem becoming a serious programmer.

Each of these examples could have been written in any language. The main point of these programs is that they both contain a loop, which is another way of saying that a particular task is being executed over and over again.

Summary

We looked at the basics of the computer, including its hardware and software. We examined the computer as an electronic machine that can recognize changes in states of "on" and "off"; these states represent the basis of the digitization of information for the computer. Programming relies on the ability to write clear, step-by-step solutions called algorithms in a language. Some languages are high level and easy to use, while other languages are more difficult to learn because they are low-level and closer to the machine's "understanding."