AS Computer Science

Algorithms & Programs an introduction

Understanding the role of the algorithm

Ian Patterson

Introduction to Problem Analysis & Program Design

What is problem analysis?

This means looking at a problem and understanding what is required in order to solve it. This understanding is necessary if a computer program is to be written to solve the problem.

TASK: Think of two examples where problems may occur if we don't have a complete understanding of the problem.

What is program design?

The programmer must try to find a solution to the given problem in a form that can be carried out by the computer. The design has the following characteristics:

- It results in a set of instructions that tell the computer what to do
- It is not written in a computer language.
- It must be able to be translated into a computer language.
- The result is known as an **Algorithm**.

TASK: What is program design?

Algorithm

An algorithm is an unambiguous set of instructions that describe how to perform a specified task. From the algorithm the instructions should be able to be translated into a computer language.

An Algorithm is the concept of a program, not the program itself.

TASK: Define clearly for yourself what the definition of an algorithm is.

How is an algorithm represented?

The algorithm must be represented in a clear, precise and unambiguous form. There are many ways of representing algorithms, some using diagrams, others using only text.

We will start by using a preferred written method called **Pseudo Code**. This is not a computer language, but it's structured in a similar way. The pseudo code algorithm should be able to be translated into a computer program in any implementation language.

Standard Constructs

The standard constructs used in algorithms & programming are sequence, assignment, selection and repetition (iteration).

A **sequence** of instructions occurs when the program instructions or statements are executed one after another.
e.g. A sequential algorithm

```
input Num1
input Num2
Sum ← Num1 + Num2
output Sum
```

Sequence is very important for algorithms as we move forward. The sequence may be thought of as the correct ordering. This is vital when our programs become more complex.

Assignment is when a variable takes on a value. e.g. An assignment statement in a wage calculation program OverTimeRate = 10.67

Sometimes we see assignment happening in a loop, for example if the count variable is initialised as 1.

```
Count=1

While Count < 12
  Input from user
  Count=Count+1
End While
```

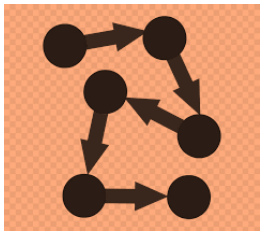
Selection occurs when an outcome depends upon various conditions. e.g. Selection statements include

```
if ... then ...
if ... then ... else ...
```

Repetition (Iteration) occurs when part of a program or algorithm may be carried out more than once, i.e. the statements form part of a loop.
e.g. Repetition (iteration) statements include
for loop
while loop

All computer solvable solutions can be written using only these constructs.

TASK: Define clearly for yourself the standard constructs.



Activity 1 – Looking at a Problem. Identifying Sequence.

The following instructions are not in the correct sequence to successfully complete the stated task. Rewrite them so that they form the correct sequence!

TASK: Put the instructions in sequence.

Process exam results
Print results
Convert marks to percentages
Assign grades to candidates
Mark papers
Calculate percentage grade boundaries
Collect answer papers

→

Process exam results

Withdraw cash from Automated Cash Point
Take cash
Remove card
Select ‘cash with receipt’ service
Insert card
Select ‘no’ to ‘another transaction’ question
Take receipt
Type in PIN
Enter amount required

→

Withdraw cash from Automated Cash Point

Activity 2 - Dry Runs & Trace Tables. Checking our Algorithm.

A DRY RUN is when an algorithm is tested on paper rather than on a computer.
The algorithm is worked through line by line, as each instruction is carried out the contents of various locations in the computer's memory is noted down along with any output. These are shown in a trace table.

Here is a dry run table for a simple algorithm.

e.g.1)

	First	Second	Third	O/P
First ← 1	1			
Second ← 2		2		
Third ← First + Second			3	
output Third				3

Notice how in the trace table only one item changes as you go from one line to the next?

e.g.2)

	Num1	Num2	O/P
Num1 ← 4	4		
Num2 ← 8		8	
Num1 ← Num1 + 2	6		
Num2 ← Num1 + Num2		14	
output Num1			6
output Num2			14

TASK: Complete the following trace tables.

1)

	Price	Vat	Cost	O/P
Price ← 100				
Vat ← 17.5				
Cost ← Price + Vat				
output Cost				

2)

	J	K	L	O/P
J ← 2				
K ← 6				
L ← J*K				
J ← J+1				
L ← J*K				
output J				
output K				
output L				

3)

	P	Q	R	O/P
P ← 6				
Q ← 8				
R ← 3				
P ← Q * R				
Q ← R*R				
output R				

4) DATA 3,9

	A	B	C	O/P
input A				
input B				
C ← A + B				
output C				

5) DATA 4,5

	L	M	N	O/P
input L				
input M				
N ← L * M				
output N				

6) DATA 4

	P	Q	R	O/P
input P				
Q ← P * 2				
R ← 3				
Q ← P * R				
P ← R*R				
Output Q				

Activity 3 - Simple Problems involving Sequence

Example: To calculate the area of a triangle

Formula: Area = Base x Height / 2

Analysis of problem:

We need to input the base length and the height – we need variables to hold these values.
We need to calculate the area using the formula.
We also need a variable to hold the answer to the calculation.
We must output this value.

Algorithm:

```
begin
input Base
input Height
Area ← Base x Height / 2
output Area
end
```

Example 2: To enter 12 monthly rainfall figures into our program, store them in our array (list) and then loop back through the array to print all the values out.

Make sure you have downloaded Python on to your PC or laptop.
Let's look at a program that allows monthly rainfall readings in a year to be entered into an array.

When generated, try and write as an algorithm.
Obviously, when you get more proficient you do the process to other way around!

The first step after entry of the readings, we'll use a loop again to print them out to screen again to verify.

Now let's implement such a program. When you write your own code use comments in the code to explain to yourself what the code is doing. We use the # symbol to start a comment. The text will be red and therefore ignored when the code is translated to machine code. My comments are done on the image in Word therefore they're in those ugly boxes.

Follow the code below, this may be the first time you've coded so keep calm. It will get easier to do, it's about practise and familiarity.

```
count=0
readings=[]
NoOfReadings = 12

while(count < NoOfReadings):
      print ('The count is:', count)
      inputvalue = input("Enter your reading: ")
      readings.append(inputvalue)
      count += 1
print ('Thank goodness. The values are stored in the array!')



count=1
for x in readings:
      print ('Reading No: ',count,' was entered as:', x , 'It is in subscript pos: ', count-1)
      count+=1

print('\n')
```

Here you can see variables which are initialised (assigned at value at run time). There is a data structure too called a list or array =[ ]

The loop iterates whilst count is less than NoOfReadings 0 to 11

Here you can see both output and input used. Also, we store the 'inputvalue' in the array

Count is assigned its new value. Can be written as count=count+1

Here is the use of a FOR loop. It is iterating through the array called readings

Here you can see the output making use of concatenation.

The FOR loop doesn't need the increment of x by us at the bottom (like in a WHILE loop). The use of assignment at the bottom is for the count value only.

Activity 4 – More challenging Problems

Have a go at each program.
Make sure you add comments to your code in Python.
Write it for yourself as an algorithm
Test with different data sizes.

Number 1 Sorting data items in an array: Ascending order
Bubble Sort

```
def bubblesortfunction(MyArray):
        moreswapstodo=True

    while moreswapstodo = True
        moreswapstodo=False
        for index in range(len(MyArray)-1):
            if MyArray [index]> MyArray [index+1]:
                moreswapstodo=True
                temp= MyArray [index]
                MyArray [index]= MyArray [index+1]
                MyArray [index+1]=temp
            end if
         end for

      end while
    return MyArray


def main():
    MyArray=[23,7,10,25,21,4,11,6,27,13]
    sortedlist = bubblesortfunction(MyArray)
    print('The list now sorted = ',sortedlist)
```

The array is passed over for sorting. Set moreswapstodo to True and therefore fall into the loop for checking. It may already be sorted? It may be unsorted? We don't know until we check!

Flip back to false in-case we never make any swaps of positions

Compare adjacent pairs. If a pair need swapping, flip the moreswapstodo back to true & swap the value positions.

Call function + send over array. The sorted array will be sent back on the return to sortedlist.

The list will be printed out.

Make sure you add comments to your code in Python.
Write it for yourself as an algorithm
Test with different data sizes.

Number 2 Searching for a data item in an array
Linear Search

```
def linearsearch(MyArray,searchvalue):
    found = False
    pointerposition = 0

    while pointerposition < len(MyArray) and found==False:
        if MyArray [pointerposition]== searchvalue:
            found=True

    pointerposition=pointerposition+1
    end while

    return found
```

Initialise variables

Loop has two conditions pointer is less than size of array and found is false still

If we match content of array with search value, flip the found to true. This will stop the loop

Increment pointer by 1 each time

True or false sent back to located

```
def main():
    MyArray=[23,1,49,8,17,22,4,9,11,44,16,27]
    searchvalue=input('What value are you seeking? : ')
    located=linearsearch(MyArray,searchvalue)

    if located = true:
        output('Your search was successful')
    else:
        output ('Your search was unsuccessful')
```

Set contents of array

Get search value

Call function + send over array and search value.
True or false will be sent back from return found to variable called located.

Make sure you add comments to your code in Python.
Write it for yourself as an algorithm
Test with different data sizes.