# Worksheet 3 Writing and interpreting algorithms

**Task 1**

1. An Internet site has the following login procedure. To access their account, a customer has to enter

* a 10-character user ID (3 attempts allowed)
* a 4-digit PIN (1 attempt allowed)
* three random characters from their password (3 attempts allowed)

 Once the user enters their user ID, a subroutine is called which looks up their file and reads the stored userID, PIN and password.

 If the user fails after 3 attempts to enter an ID which is held on the file, a message is displayed “Incorrect ID – access denied”

 Similar messages are displayed if the user fails to enter a correct PIN or password

 If all details are correct, access to the account is permitted.

 (a) Draw a hierarchy chart to show the tasks and subtasks involved in the login procedure.

 (b) Write pseudocode for a subroutine to verify the user ID.

 (c) What are the weaknesses in this login procedure? What improvements would you suggest?

**Task 2**

2. The following numbers are to be sorted into ascending order using an **insertion sort**:

15, 73, 29, 66, 35, 11, 43, 21

 (a) Show the sequence of the numbers after each pass through the insertion sort algorithm

 userNumber = [15, 73, 29, 66, 35, 11, 43, 21]

 #assume first element of array is userNumber[0]

 FOR j = 1 to LEN(userNumber)

 nextNum = userNumber[j]

 i = j – 1

 WHILE i >= 0 and userNumber [i] > nextNum

 userNumber[i + 1] = userNumber [i]

 i = i - 1

 ENDWHILE

 userNumber[i + 1] = nextNum

 ENDFOR

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **73** | **29** | **66** | **35** | **11** | **43** | **21** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

 (b) How many passes are made through the data for a dataset of n numbers?

3. Write an algorithm which will compare the length of time to sort a list of n random numbers using a bubble sort and an insertion sort.

 If you have time, write the program in a language of your choice and experiment with different values of n. Be prepared to wait a long time if you choose a number greater than 10,000!

 You can find an algorithm for a **merge sort** on the Internet and time that one too.

 Which is fastest for 10 numbers?

 Which is fastest for 100 numbers? 1000 numbers?

 If you run the program two or three times for the same number of values, why is the time taken different on each program run?