Assessment Task

Qualification national code and title	ICT50718 Diploma of Software Development
Unit/s national code/s and title/s	ICTPRG523 Apply advanced programming skills in another language

Assessment type (☑):

☐ Questioning (Oral/Written)
☐ Practical Demonstration
☐ 3rd Party Report
☒ Other – Project/Portfolio *(programs)*

Assessment Resources:

Personal computer with Python IDE (PyCharm or similar) and internet access.
Access to Blackboard shell "ICTPRG523 Advanced programming skills in another language".

Assessment Instructions:

This assessment requires you to write a program in Python that demonstrates hashing, sorting, and searching algorithms.

Due date: end of week 12.

1. Complete all the assessment tasks below.
2. Observation by your lecturer of you doing the assessment is considered part of the assessment process.
3. Submit your result and supporting documentation into the Blackboard assessment area.
4. All skills must be demonstrated to achieve a satisfactory result.

All work submitted must be your own individual effort.

Assessment Task

Qualification national code and title	ICT50718 Diploma of Software Development
Unit/s national code/s and title/s	ICTPRG523 Apply advanced programming skills in another language

Assessment Instrument:

Assessment 2: Hashing, Sorting, and Searching

You are working for a boutique software development company in Perth. The department head has asked you to write some programs to highlight the types of problems you generally solve to a new cohort of (junior) employees.

For quality assurance purposes, you may also need to debug and test your programs. You must also show the correct use of comments and other documentation (e.g., docstrings). This will help the new employees to learn to organisational requirements that apply to writing software.

Please complete all four tasks:

1. Discuss the advantages and disadvantages of three (3) sorting algorithms.
2. Write a program that creates a stem-and-leaf diagram using the data provided below:
    a. Your program must implement a sorting algorithm to sort the data into ascending order.
    b. Your program must then apply a hashing algorithm or function to the data to construct the stem-and-leaf data structure representation. Print out the data distribution.
    c. Your program must use a searching algorithm to allow the user to search for a particular value in the data set.

Sample input data:

36 29 31 125 139 131 115 105 111 40 119 47 105 57 122 109 124 115 43 120 43 27 27 32 61 37 127 29 113 121 58 114 126 53 114 96 12 127 28 42 39 113 42 18 44 18 28 48 125 107 114 34 133 45 120 30 127 31 116 146 58 109 23 105 63 27 44 105 99 41 128 121 116 125 118 44 37 113 124 37 48 127 25 109 7 31 141 46 13 27 43 117 116 27 7 68 40 31 115 124 42 128 52 71 118 117 38 27 106 33 117 116 132 104 123 35 113 122 42 117 119 32 61 37 127 29 113 121 58 114 126 53 114 96

Notice there are duplicate values in the input data (e.g., 127), so your program must be able to handle hash code collisions in the data structure.

Comment your programs and upload your evidence in compressed format into the Blackboard assessment area.