B2.5

File processing

CHAPTER ASSIGNMENT · 70 MARKS

Name _____ Class _____ No. _____ Date _____

Answer every question in the space given. Show your working. You may use the chapter page to help you.

Questions from *Computer Science for the IB Diploma*, 1st edition (2025), Paul Baumgarten, Ioana Ganea, Carl Turland (Hachette Learning), for classroom use.

Programming Exercises — File Processing (p. 388)

1. Construct a program that checks whether a given file exists (called `name.txt`). If the file does not exist, the program asks the user what their name is, prints a personalized greeting message to them and saves the person's name into `name.txt`. If the file does exist, instead of prompting for their name, it loads the name from the file and prints the personalized greeting immediately.

2. Construct a program that uses a file to store a number to represent the number of times the program has run. Every time the program runs, it should increase the number by one and save the new value.

3. Construct programming code that generates 100 random numbers with values between 1 and 70. Store those numbers into a text file, one number per line. Call the file `numb.txt`.

4. Construct code that uses the `numb.txt` file to read the numbers, identify the number of duplicate values and delete those duplicate values from the file. Output how many numbers were deleted.

5. Construct code to append to the same file the exact number of elements that were deleted in question 4, including values from 71 to 150.

6. Construct programming code to read the text file and sort the values in ascending order using an efficient bubble sort algorithm.

7. Construct a program to read the values from the text file and identify the odd numbers. Store those numbers into a new text file.

Linking Questions (p. 388)

1. Does database programming in SQL require computational thinking? (A3)

2. Why is an understanding of variables and their scope important for effective memory management in computer systems? (A1)

3. Is algorithmic efficiency relevant to machine learning, where large data sets are processed, and computational cost can be significant? (A4)

4. Are data structures, such as stacks and queues, applicable in networking algorithms for packet routing and load balancing? (A2)

5. How can graph theory be applied to packet distribution in networks? (Mathematics A&I HL)

6. How do graph algorithms and terminologies, such as vertices and edges, impact machine learning algorithms like network analysis? (A4, Mathematics A&I HL)

7. How can network traffic be used as an example or connection to programming algorithms? (A2)

8. How can programming algorithms be used to develop machine learning methods? (A4)

Exam Practice Questions — Topic B2 Programming (pp. 389–391)

1 Consider the following code that processes an input string:

```
public static void content(String text) {
    String value = "";
    int pos = 0;
    while (pos < text.length()) {
        String letter = text.substring(pos, pos + 1);
        if (letter.equals(",")) {
            System.out.println(value);
            value = "";
        } else {
            value += letter;
        }
        pos++;
    }
    if (!value.isEmpty()) {
        System.out.println(value);
    }
}
```

```
def content(text):
    value = ""
    pos = 0
    while pos < len(text):
        letter = text[pos:pos+1]
        if letter == ",":
            print(value)
            value = ""
        else:
            value = value + letter
        pos = pos + 1
    if value:
        print(value)
```

(a) Trace the algorithm for the input value of `"car,boat,ball"` by copying and completing the following table, up to the variable `pos` becoming 7. [4]

VALUE	POS	LETTER	LETTER = ","	OUTPUT

(b) Deduce the purpose of the algorithm. [3]

(c) The output is dependent on the exact format of the input. Identify two strings that would not generate the desired output. [2]

2 A group of friends play a round-robin tournament game of table tennis, where each person plays against each other player for 5 minutes at a time. The information from the tournament is recorded in parallel arrays (lists), `NAMES` and `SCORES`, examples of which are shown below.

When two people have their round against each other, each player records the points they win in their assigned row, in the column for the person they are playing against. For example, in the data set below, when Annabelle played against Jack, `SCORES[0][3]` shows that Annabelle scored 4 points, and `SCORES[3][0]` shows that Jack scored 8 points.

Index	NAMES	0	1	2	3	4	5
0	Annabelle	0	3	4	4	3	5
1	Benjamin	5	0	3	5	3	3
2	Claire	7	5	0	4	3	5
3	Jack	8	6	4	0	5	4
4	Fran	2	1	4	5	0	2
5	Mark	2	3	2	5	4	0

(a) Construct an algorithm that will declare and populate the `NAMES` array (list), based on the example data above. [2]

(b) Identify the data type of the elements in the `SCORES` array (list). [1]

(c) Identify the winning player in the game between Claire and Benjamin. [1]

(d) Construct an algorithm that will print the names of players for every round, each player's respective scores and who the winner was. If the game was a tie, output an appropriate message. For example:

`Annabelle scored 3 vs Benjamin scored 5: Benjamin won`

`Annabelle scored 4 vs Claire scored 7: Claire won` [6]

(e) Construct an algorithm that will declare two new parallel arrays (lists) to keep track of a leaderboard based on net points each player has won. Perform the calculations necessary to populate these arrays. The two parallel arrays should be `LeaderName` and `LeaderPoints`. Each player's element in the `LeaderPoints` array should be the total of points they won minus the total of points they conceded.

For example, the first two rows using the data above would be:

`Annabelle: -5` (Annabelle won 19 points but conceded 24)

`Benjamin: 1` (Benjamin won 19 points and conceded 18) [8]

(f) Construct an algorithm that will sort the parallel arrays (lists) of `LeaderName` and `LeaderPoints` in descending order of `LeaderPoints`. [6]

(g) Construct an algorithm that will save the parallel arrays (lists) of `NAMES` and `SCORES` to `names.txt` and `scores.txt` text files. [4]

3 A company has exported its sales data from a spreadsheet to a couple of text files. `NAMES.txt` contains a list of the names of its salespeople and `SALES.txt` contains a list of the total sales made by each person in the last month.

NAMES.txt	SALES.txt
Amina	23424
Carlos	42549
Emily	52488
Hao	37562
Isabella	44770

(a) Construct an algorithm that will declare two parallel arrays or lists, `NAMES` and `SALES`, open the files and load their content into the respective array or list. You may assume there is a maximum of 1000 entries in the file. [4]

(b) Construct an algorithm that will use a recursive quicksort to sort the two parallel arrays or lists by `SALES` in descending order. [8]

(c) The content of the `NAMES` array (list) was iterated over with a loop, and its elements added into an empty stack (element 0 was the first to be added to the stack). What can be said about the order of the `NAMES` that will be popped off the stack in respect to their sales? [3]

4 (a) Define the term "queue". [2]

(b) Describe an application that uses a queue in a computer system. [2]

5 State the efficiency of the quicksort algorithm in Big O notation. [1]

6 Outline two uses of a stack in a computer system. [4]

7 Outline the differences between storing a queue in a linear form as opposed to a circular form. [4]

8 State which data type is best for storing a telephone number (e.g. `00352 661 008 990`) and give two reasons. [3]

9 Arrays or linked lists can be used to implement stacks and queues.

(a) Describe the advantage of using an array to implement a stack or a queue. [1]

(b) Describe the advantage of using a linked list to implement a stack or a queue. [1]

10 The following list of numbers needs to be put into ascending order: `3, 1, 7, 2, 4, 1, 6`. State the list that would be obtained after two iterations of a selection sort. [1]

11 Define the term "recursion". [1]

12 Construct a program to calculate the sum of numbers from 1 to n using recursion. [4]

13 State one advantage and one disadvantage of recursion vs iteration. [2]

14 Consider the code shown below.

```
public static void manipulate(int n, int[] a) {
    a[0] = 15;
}
public static void main(String[] args) {
    int[] a = {0, 1};
    manipulate(3, a);
    for (int i = 0; i < a.length; i++) {
        System.out.println(a[i]);
    }
}
```

```
def manipulate(n, a):
    a[0] = 15
a = [0, 1]
manipulate(3, a)
print(a)
```

(a) State the scope of the variable `n`. [1]

(b) An integer array (list) called `b` is initialized with the values `{11, 12, 13, 14}`. The procedure `manipulate` is called again with the parameters `14` and `b`. Explain why the assignment `a[0] = 15` in the body of the procedure changes the values stored in the array (list) `b`. [3]

15 Consider the following data structure:

```
int[] a = {1, 2, 3, 4, 5};
```

```
a = [1, 2, 3, 4, 5]
```

(a) State the name of the data structure. [1]

(b) Outline how the data value 3 can be directly accessed. [1]

End of assignment · Check your answers against the chapter at gurujazzy.com/subjects/ib-computer-science/b2-5-file-processing/