Langara College

**Department of Computing Science and Information Systems**

**CPSC 1150** – **Program Design** **Final Exam**

**Spring 2021**

**Time: 2 hours (including this part and MCQs)**

**Total Marks: 100**

**Name: Jay Seung Yeon Lee ID: 100357736**

**Instructions:**

1. Download the file **Final.zip** from BrightSpace and save all your files inside this folder **Final**
2. For the algorithm design question, enter your answer below the question.
3. For all the programming problems, save the programs as the names provided. Add comments where necessary, but full internal documentation is not required. NO external documentation is required.
4. At the end of the test, zip the folder **Final** and upload it to BrightSpace.
5. Since the programs will be tested under SciTE, make sure they can be compiled and executed underSciTE.
6. By taking this test, you declare you will not cheat.
7. You may not collaborate with anyone.
8. You may not ask for help from anyone.
9. You may not use any library functions and/or knowledge that are not covered in this course.
10. You may use the Internet, but you may not search for solutions.
11. Multiple Choice Questions (20 marks)

The MCQ questions are posted under Assessments->Quizzes

1. Algorithm design (10 marks)

Develop **a low-level algorithm** thatcreates an array of strings and swaps the order of values in the array in a pairwise fashion. Your algorithm should swap the order of the first two values, and then swap the order of the next two, and so on, until all the pairs are swapped.

For example, if the array initially stores these values:

{"four", "scores", "and", "seven", "years", "ago"};

Your algorithm should switch the first pair ("four", "score"), the second pair ("and", "seven") and the third pair ("years", "ago"), to yield the following new array:

{"scores", "four", "seven", "and", "ago", "years"}

If there are an odd number of values in the array, the final element is not moved. For example, if the original array contains:

{"to", "be", "or", "not", "to", "be", "hamlet"}

It still swaps pairs of values, but the final value ("hamlet") will not be moved, yielding the following new array:

{"be", "to", "not", "or", "be", "to", "hamlet"}

Answer for Question 2 Algorithm design:

START

SET temp to “”;

FOR int j = 0; j < array.length -1; j+=2

IF array.length%2 == 0

IF j == array.length -1

Break

END IF

END IF

SET temp to array[i+1]

SET array[i+1] to array[i]

SET array[i] to temp

END FOR

END

// Below is the program for the algorithm

String temp = "";

for(int i = 0; i < array.length-1; i+=2){

if(array.length % 2 == 0){

if(i == array.length-1){

break;

}

}

temp = array[i+1];

array[i+1] = array[i];

array[i] = temp;

}

1. Programming (70 marks)
2. String processing (**CountEachVowel.java**) (10 marks**)**

Write a method that returns the count of each vowel (a, e, i, o, u, including both uppercase and lowercase) in a given sentence. In the main, make up a sentence, call the method above, and display the counts.

If you are given a sentence, "How are you today?"

your program should display the following result:

a: 2

e: 1

i: 0

o: 3

u: 1

1. 2D array (**MostPopularTutors.java**) (25 marks)

The table below gives the number of students of seven tutors tutored over the period of one week.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Bob** | **Alice** | **Mike** | **Jane** | **Justin** | **Paul** | **Dan** |
| **Monday** | 13 | 15 | 12 | 12 | 2 | 12 | 2 |
| **Tuesday** | 8 | 5 | 14 | 8 | 8 | 4 | 8 |
| **Wednesday** | 3 | 11 | 1 | 2 | 6 | 13 | 9 |
| **Thursday** | 3 | 12 | 4 | 9 | 15 | 15 | 10 |
| **Friday** | 6 | 3 | 12 | 14 | 2 | 2 | 15 |

1. (20 marks) Write a method that returns all the most popular tutors (maybe more than one) of the week.
2. (5 marks) Write the main method to call the method above and display the result

**Note**: You must use loops wherever possible.

// Store the number of students for the tutors in a 2D array

int[][] countOfStudents = {{13, 15, 12, 12, 2, 12, 2},

{8, 5, 14, 8, 8, 4, 8},

{3, 11, 1, 2, 6, 13, 9},

{3, 12, 4, 9, 15, 15, 10},

{6, 3, 12, 14, 2, 2, 15}

};

Based on the data given, your program should display the following result:

The most popular tutor(s): Alice Paul

1. Five stones game (**FiveStones.java**) (35 marks)
   1. The game is played between two players who alternate to place x and o stones on the vacant squares on a grid with 9×9 squares.
   2. Once placed on the grid, stones cannot be moved or removed from the grid.
   3. The player who first places five stones on one line (horizontally, vertically, or diagonally) wins the game.
   4. If all the squares are occupied and there is still no winner, then the game ends with a tie.
   5. Assume that the two players play from a single computer and from a single interface.

**Your tasks:**

[10 marks] Draw the top-down design diagram for playing the five stones game.

**You can draw the top-down diagram using a drawing tool, or draw it on a piece of paper, take a picture, and insert it here or save it in the submission folder**

[2 marks] Based on the code given, display the grid wherever necessary.

[23 marks] Write code to get valid plays from the players and place the stones in the squares until the players do not want to continue.

**Rules for a valid play:**

1. A valid play cannot be out of the grid
2. A valid play must be in a vacant square
3. A valid play type must either x or o
4. A valid play type cannot be the same as the type just played

**Notes**:

1. For implementation, you do not need to check if there is a winner or a tie.
2. You can assume that there is always at least one vacant square on the grid.
3. You can assume that the players always enter lowercase letters.
4. You must use methods wherever possible.