SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

ICT133

Examination – July Semester 2021

Structured Programming

Thursday, 18 November 2021 **1:00 pm – 3:00 pm**

Time allowed: 2 hours

INSTRUCTIONS TO STUDENTS:

1. This examination contains **FOUR (4)** questions and comprises **SIX (6,)** printed pages (including cover page).
2. You must answer **all** questions.
3. All answers must be written in the answer book.
4. This is **an open-book** examination.
5. You can use ONLY built-in functions, built-in types and packages covered in this course, e.g., math, random. Do not write classes.

At the end of the examination
Please ensure that you have written your examination number on each answer book used.

Failure to do so will mean that your work cannot be identified.

If you have used more than one answer book, please tie them together with the string provided.

THE UNIVERSITY RESERVES THE RIGHT NOT TO MARK YOUR SCRIPT IF YOU FAIL TO FOLLOW THESE INSTRUCTIONS.

Answer all questions. (Total 100 marks)

Question 1

Describe and apply control structures based on computational logic.

(a) Given the following code segment:

```
if x > y:
   if z < y:
      print(y)
   elif x < z:
      print(x)
   else:
      print(z)
else:
   if y < z:
       print(y)
   elif x < z:
       print(z)
   else:
       print(x)
```

(i) What is the output if the values of x, y and z are 1, 3, 2?

(2 marks)

(ii) What does the code segment do for different values of x, y, and z?

(3 marks)

(b) Write and express a program that converts a 24 hour clock time to a 12 hour clock time. For example, 2359 is 11.59PM and 0910 is 9.10AM. Input is a string representing the 24 hour clock. No validation is required.
Two sample output are as follows:

Sample 1

```
Enter 24 hour time: 1305
Time in 12 hour format is 1.05PM
```

Sample 2

```
Enter 24 hour time: 0010
Time in 12 hour format is 12.10AM
```

(10 marks)

(c) Write a function `validateInput(time)` that has a string parameter. The function returns `True` if the time parameter is a valid 24 hour time, and `False` otherwise. A valid 24 hour time satisfies the following conditions:

- length is 4
- contains all digits
- the first 2 digits that represent the hour must be between 0 and 23 inclusive
- the last 2 digits that represent the minutes must be between 0 and 59 inclusive

(10 marks)

Question 2

Interpret and solve computational problems using structured programming.

(a) What is the output of the following code segment?

```
x = 1
y = 2
for n in range(5):
   z = x + y
   x = y
   y = z
   print(z, end=' ')
```

(5 marks)

(b) The following function `replace` has a string parameter and replaces all occurrences of `char1` with `char2`. Both `char1` and `char2` are single characters.

```
1. def replace(string, char1, char2):
2.     for n in range(len(string)):
3.         temp = ''
4.         if string[n] == char1:
5.            temp = char2 + temp
6.         return temp
7.
8. print(replace('ajax', 'a', 'A'))
```

The function is tested with the statement in line 8. Examine the code and answer the following questions.

(i) How many times will the for loop in line 2 be executed?

(2 marks)

(ii) What output will be displayed?

(2 marks)

(iii) Rewrite the function to fix all logic errors so that the output will display as `AjAx`.

(6 marks)

(c) Write and employ a function `index(string, substring)` that has 2 string parameters.

You are not allowed to use the inbuilt string methods `index()`, `find()` in Python, and you may not convert string to a list.

The `index()` function returns the starting index position of a matching substring in the string. If there is no match, return -1. E.g. if the string is 'vaccination', the following cases will return values as:

- substring ‘a’ returns index 1, the first ‘a’ found in the string
- substring ‘cc’ returns index 2, the start index of the string that matches the substring
- substring ‘ace’ returns -1, as there is no match
- substring ‘vaccinations’ returns -1

Assume that the string and substring are all lowercase letters.

(10 marks)

Question 3

Apply data structures to solve computational problems.

(a) Develop and write a function `count(numlist)` that has a list of integers as parameter. The function returns a count of the occurrences of each number in the list as a dictionary. E.g. given a list of numbers:

```
numlist = [2, 1, 3, 6, 2, 1, 4, 1, 2, 6, 5, 1, 1, 2]
```

the function returns `{2: 4, 1: 5, 3: 1, 6: 2, 4: 1, 5: 1}`, since the value 2 occurs 4 times, the value 1 occurs 5 times etc. If `numlist` is empty, an empty dictionary is returned.

(13 marks)

(b) Write a function `mode(numlist)` that has a list of integers as parameter. Mode is the value that occurs the most times in a list. The function returns the number in the list which occurs most frequently. E.g. given the list in part Q3(a), the function returns a list with value [1], since 1 occurs the most time. It is possible that there may be more than one mode. In this case, return the mode as a list, as in [2, 5], assuming 2 and 5 occurred the most time in the list. Make use of the function in part Q3(a).

(12 marks)

Question 4

Apply data structures to store and process information. Employ structured programming principles and solve computational problems.

A university admits students based on the total points for 4 subject grades. The applicants’ grades are stored in a file. A program reads the applicants grades from a file and calculates the points based on the grades and prints a summary report on the admission eligibility. The points awarded based on the grade is given by the following table:

Grade	Points
A	20
B	16
C	12
D	10
E	6

Table Q4 Grade point table

Total points of 50 and above are considered for admission. The applicant file consists of student id, and 4 subject grades, one student per line. Refer to appendix A for the file contents.

The program consists of functions as follows:

(a) Write a function `calculatePoints(grades)` that has a grades parameter. The grades parameter is a list of grades, as in `['A', 'B', 'C', 'C'].` The function returns the total points based on the grade point table. No selection statements are allowed in this function.

(6 marks)

(b) Write a function `readFromFile(filename)` that reads applicants grades from a filename parameter. The function returns a dictionary in the following format:

```
applicants={'S1':['A','A','A','C'], 'S2':['B','B','C','C']}
```

(only 2 shown). The key is student id, and the value is a list of grades. The grades should be stripped of trailing blanks and \n characters.

(8 marks)

(c) Write a function `printEligibilityReport(applicants)` that has a dictionary of applicants as parameter. The function prints a report of the points attained for each student and their eligibility for admission to the university. Students with total points 50 or more are admitted. A sample report is as follows:

```
S1 A,A,A,C 72 pts Admitted
S2 B,B,C,C 56 pts Admitted
S3 D,D,A,E 46 pts Rejected
S4 B,C,D,A 58 pts Admitted

3 admitted, 1 rejected.
```

(7 marks)

(d) Write a main function that reads in the file and prints the eligibility report.

(4 marks)

Appendix A

applicants.txt

```
S1, A, A, A, C
S2, B, B, C, C
S3, D, D, A, E
S4, B, C, D, A
```

----- END OF PAPER -----