RIVER VALLEY HIGH SCHOOL
General Certificate of Education Advanced Level
Higher 2
Preliminary Examination

COMPUTING

9569/01

Paper 1

14 Sep 2021
3 hours

Additional Materials: Answer Paper

READ THESE INSTRUCTIONS FIRST

Write your center number, index number and name on all the work you hand in.
Write in dark blue or black pen on both sides of the paper.
You may use an HB pencil for any diagrams, graphs, tables or rough working.
Do not use staples, paper clips, glue or correction fluid.
DO **NOT** WRITE IN ANY BARCODES.

If you need additional answer paper ask the invigilator for a continuation booklet.

Answer **all** questions.
Approved calculators are allowed.

The number of marks is given in brackets [ ] at the end of each question or part question.
The total number of marks for this paper is 100.

This document consists of **9** printed pages.

Answer **all** questions.

1. Draw a reduced decision table based on the following conditions regarding how John should go to school. [5]

- If it is a Monday, John always takes his dad's car to school if he does not oversleep.
- If John oversleeps, he always takes Taxi to school.
- Otherwise, if it is a rainy day, John takes Taxi to school. If not, by MRT.

2. The recursive function below helps to check if a password string `pw` satisfies certain requirements. The meaning of the function parameters `pw`, `digits`, `upper_l`, `lower_l` and `length` are password string, minimum number of digits, minimum number of uppercase letters, minimum number of lowercase letters and minimum length of the password respectively.

```
def check_pw(pw, digits, upper_l, lower_l, length):

    if len(pw) == 0:
        return digits < 1 and upper_l < 1 and lower_l < 1 and length < 1
    else:
        char = pw[0]
        if char.isdigit():
            return check_pw(pw[1:], digits-1, upper_l, lower_l, length-1)
        elif char.isalpha():
            if char.isupper():
                return check_pw(pw[1:], digits, upper_l-1, lower_l, length-1)
            else:
                return check_pw(pw[1:], digits, upper_l, lower_l-1, length-1)
        else:
            return False
```

a) State the values of all arguments in each recursive function call when the following code is executed. Then, state the value that the function returns.

```
>>> check_pw("SP500", 3, 1, 1, 5)
```

[3]

The function in **2a)** is rewritten in such a way that string slicing on the password string `pw` is removed. A new function parameter `i` is added to help the recursive function to keep track of the position in `pw` in which it is currently checking.

```
def check_pw(pw, i, digits, upper_l, lower_l, length):
    if A:
        return digits < 1 and upper_l < 1 and lower_l < 1 and length < 1
    else:
        B
        if char.isdigit():
            return check_pw(pw, C, digits-1, upper_l, lower_l, length-1)
        elif char.isalpha():
            if char.isupper():
                return check_pw(pw, C, digits, upper_l-1, lower_l, length-1)
            else:
                return check_pw(pw, C, digits, upper_l, lower_l-1, length-1)
        else:
            return False
```

b) State the code in **A**, **B** and **C**. [3]

c) Describe clearly or write in Python the modification needed for the function `check_pw()` to also display a suggestion of a new password if the password requirements are not met. For example, if `pw` does not have enough digits, it will append digits to `pw` so that it can satisfy the requirement.

For example,

```
>>> check_pw("WoBeiShiGeDa", 0, 2, 6, 7, 15)
Suggested password: WoBeiShiGeDa33Uxxx
False
```

`"33Uxxx"` is added to `"WoBeiShiGeDa"` so that the password passes the requirements. [4]

3. The following recursive procedure is created to encode a character `char` based on a `shift` value. For example, if letter `"a"` is shifted by `3`, it will become letter `"d"`; and if shifted by `-3`, it will become letter `"x"`.
The `ORD` and `CHR` function will help to convert the character to its ASCII value and vice versa.

```
01 PROCEDURE ENCODE_CHAR(char: STRING, shift: INT):
02     IF shift == 0:
03         RETURN char
04     ELSE IF shift > 0:
05         DECLARE new_char: STRING
06         new_char = CHR((ORD(char) + 1) % 26)
07         shift -= 1
08         RETURN ENCODE_CHAR(new_char, shift)
09     ELSE:
10         Shift += 26
11         RETURN ENCODE_CHAR(char, shift)
12     END IF
13 END PROCEDURE
```

a) Identify one error from the above code, state the type of the error, including its definition and explain how the errors can be fixed. [2]

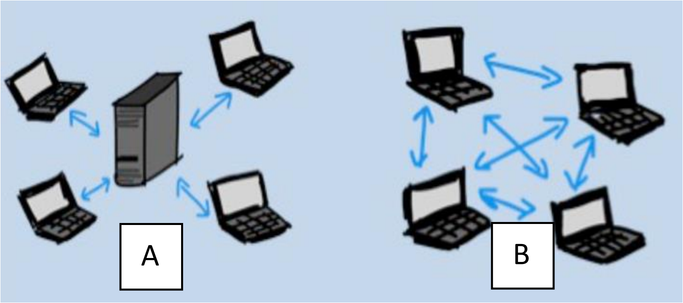
b) Assume the above error has been fixed. Copy the following trace table to your answer booklet. State the line number each time one of the return statements is called and complete it based on the following function call.

`ENCODE_CHAR("y", -24)` [3]

Line No.	**char**	**shift**	**new_char**

4. Answer all questions.

a) State the network architecture model of A and B as shown in the diagram below. [1]



b) State an advantage of model A over B in the diagram above. [1]

c) State and explain if each of the following statements are correct. [6]

- “One of the functionalities of the DNS is that different users can simultaneously receive different IP translations for the same domain name.”
- “The 4 top layers of TCP/IP model are application, internet, data link and physical.”
- “The internet layer is not responsible for reliable transmission. It makes no guarantees about the proper arrival of packets.”
- “2C:54:91:G8:F9:E3 is a valid MAC address.”
- “2001:0db8:0001:0ab9:C0A8:0102 is a valid IPv6 address.”
- “The internet and the World Wide Web are the same thing.”

d) State the purpose of HTTP and explain how the protocol works. [3]

e) Explain packet switching. [1]

f) State an ethical issue related to artificial intelligence. [1]

g) State the purpose of defining the code of conduct for computer use. [2]

h) State a difference between data validation and data verification. [1]

5. The implementation of a Binary Search Tree (BST) using three 1D arrays is shown below.

Each unused node that are not in the logical BST is initially connected in a singly linked list manner using the `leftPtr` array. The first position of this linked list is indicated by the variable `nextFree`.

When a piece of data is inserted into the BST, a node will be disconnected from the linked list and added to the logical BST. The root of this logical BST is indicated by the variable `root`. The logical structure of the BST is managed by `leftPtr` and `rightPtr` which are the positions of the left and right child of the node respectively.

Below is an illustration for such BST with a 0-based index array.

`root`	7
`nextFree`	0

Array Index	0	1	2	3	4	5	6	7
`Data`	-	7	10	6	1	4	2	9
`leftPtr`	-1	-1	-1	5	-1	6	4	3
`rightPtr`	-1	-1	-1	1	-1	-1	-1	2

a) Draw the logical BST at this point of time. [1]

b) State the post order traversal of the BST [1]

c) State the values of `root`, `nextFree` and the values in the arrays `data`, `leftPtr` and `rightPtr` after the **each** of the following BST operations are executed sequentially. [3]

- `Add 8`
- `Recursive Delete 6`

d) State an advantage of BST over Hash table. [1]

e) Explain what can make a Hash Table Search inefficient besides a bad hash function and how to overcome it. [2]

f) State 2 characteristics of a good hash function. [2]

6. Study the following sorting code carefully.

```
def sort(lst):
    if len(lst)<=1:
        return lst
    else:
        pivot = lst[0]
        smaller = []
        larger = []
        for i in range (1, len(lst)):
            if lst[i] < pivot:
                smaller.append(lst[i])
            else:
                larger.append(lst[i])
        return sort (smaller) + [pivot,] + sort (larger)
```

a) State the name of the above sorting algorithm. [1]

b) Explain why the above sorting algorithm is inefficient when it is used on a nearly sorted array. [1]

c) Explain how you can modify the code to improve the efficiency. [2]

Bubble sort and insertion sort are both used to sort a nearly sorted integer array of size 1000 and there are only 5 integers in the array that are not in the correct position.

d) State and explain why insertion sort generally perform better than bubble sort for nearly sorted array. [2]

e) Draw a flow chart for the algorithm described below. [4]

Given an integer `k` and an array `arr[]` representing the destination floors for `n` people waiting currently at the ground floor and `k` is the capacity of the elevator. It takes 1 unit time for the elevator to reach any consecutive floor from the current floor. The algorithm finds the minimal time taken to get all the people to their destination floor and then return to the ground floor.

```
def minTime(n, k, arr) :
    # Sort in descending order
    arr.sort(reverse = True)
    minTime = 0
    # Iterate through the groups
    for i in range(0, n, k) :
        # Update the time taken for
        # each group
        minTime += (2 * arr[i])
    # Return the total time taken
    return minTime
```

7. A relational database is created to store data about contractors engaging workers to perform renovation jobs.

The database designers are told that:

- each contractor can recruit different workers to perform various jobs.
- each worker can have skills to perform different jobs.
- each job can have different levels of skills "A", "B" or "C" and their hourly rate is calculated based on their skill level for the job.

A first attempt is represented by the following table:

Contractor Name	Contractor Contact	Worker Name	Job	Skill Level	Hourly Rate	Date	Starting Time	Ending Time
Su Ming De	94190731	Tan Yong Quan	Carpentry	A	40	20200801	0900	1300
		Goh Yi Xi	Carpentry	B	35	20200801	0800	1700
		Goh Yi Xi	Electrical	A	42	20200801	0900	1600
		Chong Jun Jie	Ceiling	B	23	20200801	1000	1200
Qin Kai Hui	95967027	Tung De Ming	Waterproofing	A	30	20200802	0900	1300
		Tan Yong Quan	Carpentry	A	40	20200802	1200	1800
		Goh Siew Ming	Waterproofing	A	30	20200802	0900	1300
Shen Rui Min	98022248	Tung De Ming	Waterproofing	A	30	20200803	0900	1300
		Tan Yong Quan	Carpentry	A	40	20200803	1000	1500
		Goh Siew Ming	Ceiling	B	23	20200803	0900	1300

a) Explain why this table is not in first normalized form. [2]

The following is an attempt to reduce data redundancy:

Contactor

Contractor ID	Contractor Name	Contractor Contact
1	Su Ming De	94190731
2	Qin Kai Hui	95967027
3	Shen Rui Min	98022248

Worker

Worker ID	Job	Worker Name	Skill Level	Hourly Rate
1	Ceiling	Chong Jun Jie	B	23
2	Ceiling	Goh Siew Ming	B	23
2	Waterproofing	Goh Siew Ming	A	30
3	Carpentry	Goh Yi Xi	B	35
3	Electrical	Goh Yi Xi	A	42
4	Carpentry	Tan Yong Quan	A	40
5	Waterproofing	Tung De Ming	A	30

Contract Record

Contractor ID	Worker ID	Job	Date	Starting Time	Ending Time
1	4	Carpentry	20200801	0900	1300
1	3	Carpentry	20200801	0800	1200
1	3	Electrical	20200801	1300	1600
1	1	Ceiling	20200801	1000	1200
2	5	Waterproofing	20200802	0900	1300
2	4	Carpentry	20200802	1200	1800
2	2	Waterproofing	20200802	0900	1300
3	5	Waterproofing	20200803	0900	1300
3	4	Carpentry	20200803	1000	1500
3	2	Ceiling	20200803	0900	1300

b) Explain what a composite key is. [1]

c) State a suitable primary key for Worker table and explain why the table is not in second normal form. [3]

d) A table description can be expressed as:
`TableName (Attribute1, Attribute2, Attribute3, …)`

The primary keys are indicated using a solid underline and foreign keys are indicated by using a dashed underline.

Write table descriptions for the required tables in the database so that they are in third normal form (3NF). [6]

e) Create an entity-relationship (ER) diagram showing the degree of all relations. [3]

f) Using the above example, elaborate why a relational database model has advantage in maintaining data integrity over a flat file system. [3]

g) The homeowner would like to know a schedule of the renovation jobs performed to their house. They are **NOT** interested in knowing the exact worker's name. Write an SQL query to output the **contractor's name**, **worker's job**, **worker's skill level** and **date** based on the contractor's name `"Su Ming De".` The output is to be in the ascending order based on the date of job performed. [5]

8. You are also to design an Object-Oriented solution for the above-mentioned project. Both contractor and workers are to create a `User` account on the platform, with details such as `user_id`, `password` and `gender`.

The contractors will have to register their company details such as company `name` and `address`, while the workers need to register their bank `account` number.

a) Draw a class diagram, with base class **User**, showing:

- appropriate sub-classes,
- inheritance,
- the properties required,
- appropriate methods, including but not limited to the **constructor** methods, and at least **one** pair of **'get'** and **'set'** methods for each class,
- circle the polymorphed methods.

[6]

b) Using the above example, state the definition of inheritance and explain its purpose/advantage in object-oriented programing. [3]

The platform hopes to expand its function to allow register of homeowner accounts. The homeowners can view which are the workers came to their home address for renovation work on the date/time specified by the contractors.

c) State how this would affect the **class**, **properties** and **methods** in the current example. [3]

d) State how this would affect the tables, attributes and relationships of the relational database stated in **7(d)** and **(e)**. [3]

e) Explain how NoSQL addresses shortcomings of relational databases. [4]

Some homeowners request to have access to the hourly rate and personal contact of renovation workers.

f) From the perspective of the company, explain to the homeowners how such a feature is against the data protection obligations stated in the Personal Data Protection Act (PDPA). [2]

END