一、選擇題 **(答案請畫填於電腦卡)**

1. Regular expressions. (3 points)

A string over the alphabet {a, b, A, B } is characterized by

(1) starts with a lowercase letter; (2) no two consecutive uppercase letters; (3) ends with a lowercase letter;

The examples of matches: *aaaa, aaaBa, aBaaAaa, baBabAaAbAbb*, and *a*.

The examples of NOT matches: *Baaaa, aaaaB, ABABBBA, aBAAaba,* and *B*.

Which regular expression below matches the strings?

(A) (a|b) (.*) (a|b)*

(B) (a|b) (a|b)* ((A|B) (a|b) (a|b)*)*

(C) ((a|b)* (A|B) (a|b))*

(D) (a|b) ((a|b)* (A|B) (a|b))* (a|b)

2. Convert 126 from decimal to hexadecimal

(A) 7E

(B) 3F

(C) 7F

(D) 3E

3. Let $F_k$ denote the kth Fibonacci number. Reminder: the Fibonacci numbers are defined by $F_0 = F_1 = 1$, and $F_k = F_{k-1} + F_{k-2}$ for $k \geq 2$. Consider the following variation on merge sort, which assumes that the number of elements in its list argument *L* is a Fibonacci number $F_k$.

```
algorithm FibMergesort(L)
{L is a list of items from a totally ordered set, whose length is a Fibonacci number F_k}
if L contains only 1 element, then return L
else
divide L into L_1 (the first F_{k-1} items) and L_2 (the remaining F_{k-2} items)
sortedL_1 := FibMergesort(L_1)
sortedL_2 := FibMergesort(L_2)
sortedL := Merge(sortedL_1,sortedL_2)
return sortedL
```

Assuming that the "divide" step in FibMergesort takes constant time (no comparisons) and Merge behaves as described in the lecture notes, identify which of the following expressions most closely matches the total number of comparisons performed by FibMergesort when initially given a list of $F_k$ elements.

(A) O(k log k)

(B) O(k2)

(C) O(k $F_k$)

(D) O($F_k$ log k)

(E) O(F2k)

4. Please answer the 1's complement and 2's complement of $(10110110)_2$

(A) 01001001 and 01001010

(B) 01001010 and 01001001

(C) 01001001 and 10110110

(D) 10110111 and 10111010

5. Please select the correct statements

(i) The fetch-and-execute cycle is the way the CPU of a computer executes programs written in machine language.

(ii) The execute cycle is done in ALU of CPU.

(iii) An interpreter translates one instruction at a time, and then executes that instruction immediately.

(iv) Java is a platform-independent language.

Answers:

(A)(i) and (iii); (B) (i) and (iv); (C) (i), (ii), and (iv); (D) (i), (ii), (iii) and (iv)

6. Which two statements describe the IP address 10.16.3.65/23?

(i) The subnet address is 10.16.3.0 255.255.254.0.

(ii) The lowest host address in the subnet is 10.16.2.1 255.255.254.0.

(iii)The last valid host address in the subnet is 10.16.2.254 255.255.254.0.

(iv)The broadcast address of the subnet is 10.16.3.255 255.255.254.0.

Answers:

(A)(i) and (iii); (B) (i) and (iv); (C) (i), (ii), and (iv); (D) (ii), (iii) and (iv)

7. Which of the following services use TCP?

(i) DHCP
(ii) SMTP
(iii)HTTP
(iv)TFTP
(v) FTP

Answers:
(A)(i) and (ii)
(B)(i), (iii), and (v)
(C)(ii), (iii), and (v)
(D)(i), (ii), and (iv)

8. Process aging is:
(A) Computing the next CPU burst time via a weighted exponential average of previous bursts.
(B) The measurement of elapsed CPU time during a process' execution.
(C) Boosting a process' priority temporarily to get it scheduled to run.
(D) Giving a process a longer quantum as it gets older.

9. Which scheduler gives each process an equal share of the CPU?
(A) Round robin.
(B) Shortest remaining time first.
(C) Priority.
(D) Multilevel feedback queues

10. Hadoop is a framework that works with a variety of related tools. Common cohorts include:
(A) MapReduce, Hive, and Hbase
(B) MapReduce, MySQL, and Good Apps
(C) MapReduce, Hummer, and Iguana
(D) MapReduce, Heron, and Trumpet

11. What is the minimum number of nodes in a full binary tree with depth 3?
(A) 3
(B) 4
(C) 8
(D) 11
(E) 15

12. Which of the following statement(s) is (are) true?
(i) Smartphone is able to connect to the Internet via 3G telcom network.
(ii) Smartphone is able to connect to the Internet via WiFi.
(iii) WiFi is another name for 3G telcom network.
Answers:
(A) (i) only.
(B) (ii) only.
(C) (i) and (ii) only.
(D) (i), (ii) and (iii).

13. The following table gives approximate running times for a program with N inputs, for various values of N.

| N | time |
|---|---|
| 1000 | 10 seconds |
| 2000 | 40 seconds |
| 5000 | 4 minutes |
| 10,000 | 16 minutes |

Which of the following best describes the likely running time of this program for N = 100,000?
(A) A few minutes
(B) A few hours
(C) A day
(D) A month
(E) A few years

14. The following code is written as a dynamically-typed, object-oriented language. The function assumes that it gets an array as its 'arr' parameter.

```
def fun_print(arr) {
  if (arr.length() > 1) {
    fun_print(arr[1..arr.length-1])
  }

  if arr[0].type_of(Integer) {
    println(arr[0])
  elsif arr[0].type_of(Array)
    fun_print(arr[0])
  }
}
```

What would you expect this function to output if called as follows:

```
  fun_print( [[[10, 2, 37]], 42, [5, 63]] )
```

(A) 10 42 5 2 63 37
(B) 5 63 42 10 2 37
(C) 10 2 37 42 5 63
(D) 10 2 37 5 63 42
(E) 63 5 42 37 2 10
(F) 42 5 63 10 2 37

15. The following code is written in a statically typed, object-oriented language. It uses a class called LinkedList that implements a singly linked list of instances of class Node. Calling `head()` on an instance of LinkedList returns the first Node in the list.

```
LinkedList linkedList = new LinkedList();
linkedList.add( new Node("A"));
linkedList.add( new Node("B"));
linkedList.add( new Node("C"));
linkedList.add( new Node("D"));
linkedList.add( new Node("E"));

Node current = linkedList.head();
Node bar = linkedList.head();
int foo = 0;

do {
  foo++;
  if(foo%2 == 0){
    bar = bar.next();
  }
  current = current.next();
} while(current != null)

System.out.println("foo is: " + foo);
System.out.println("bar is: " + bar);
```

What is printed by this code?

(A)
```
foo is: 4
bar is: C
```
(B)
```
foo is: 5
bar is: E
```
(C)
```
foo is: 2
bar is: B
```
(D)
```
foo is: 5
bar is: C
```
(E)
```
foo is: 4
bar is: E
```
(F)
```
foo is: 2
bar is: D
```

To answer the next three questions, pick from the following data structures:

1. Hash Table
2. Array
3. Stack
4. Linked List
5. Binary Tree
6. Queue

16. Which of the above data structures do we usually expect to be *indexed*? (i.e., you could refer to any of its elements by an index value):

(A) 2, 3, and 5
(B) 4 and 6
(C) 1, 2, and 6
(D) 3, 4, and 5
(E) 2, 3, and 5
(F) 1 and 2

17. Which of these structures would be usually described as first-in-first-out (FIFO)?

(A) 1
(B) 2
(C) 3
(D) 4
(E) 5
(F) 6

18. Which of these data structures is most appropriate to store a very large telephone directory of names and phone numbers, where we want to search for a person's number given an exact full name?

(A) 1
(B) 2
(C) 3
(D) 4
(E) 5
(F) 6

19. Which of these data structures is most appropriate to store a sorted dictionary of English language words and definitions? Users should be able to either search directly for a word or browse sequentially for words in alphabetical order.

(A) 1
(B) 2
(C) 3
(D) 4
(E) 5
(F) 6

20. Which of these data structures is typically used by compilers and interpreters to implement recursive functions?

(A) 1　　(D) 4
(B) 2　　(E) 5
(C) 3　　(F) 6

二、問答題(**請作答於答案卷**)

1. Normalization of data tables (10%)

   The following data shows details of projects and the employees who are placed on those projects along with their job titles and the hours worked by each employee on each project. (10%)

| PrjNo | PrjName | EmpNo | EmpName | JobTitle | NT/hr | Hrs |
|---|---|---|---|---|---|---|
| 101 | Road | A11 | J. Lin | Engineer | 1000 | 50 |
|  |  | B21 | K. Lee | Driver | 1200 | 50 |
|  |  | C31 | Y. Cheng | Sweeper | 2000 | 40 |
| 302 | Bridge | A31 | C. Lai | Steeler | 5000 | 25 |
|  |  | B42 | J. In | Mounter | 4000 | 30 |
|  |  | B51 | P. Zhao | Puncher | 3000 | 20 |
| 405 | Floor | C102 | B. Ko | Paver | 800 | 100 |
|  |  | D302 | Q. Ray | Steeler | 900 | 20 |
|  |  | P502 | B. Yuan | Engineer | 600 | 30 |

Identify any repeating groups in the above representation and describe how they can be removed to produce a relation in first normal form. Identify the key attributes of the resultant relation. **Normalize the data showing functional dependencies and how you progress from 1NF through 2NF to a set of 3NF relations.** At each stage show the primary key and any foreign keys of each relation and state assumptions that you make about any of the relationships between the columns of data.

2. Entity-Relation Diagram (10%)

   Micro loans have been gaining popularity in developing countries.　Typically, the loans will be used to finance startup or development of the borrower's company, so that there is a realistic chance for repayment. The idea is to bring venture lenders together using information technology.

In this problem, you will create an E-R diagram that describes the information necessary to manage micro loans according to the following business rules.

(1) Each borrower and lender must be registered with information about name and address.

(2) A loan starts with a loan request, which contains information about when the loan should at latest be granted, the total amount of the loan, and how long the payback period is. It also contains the information regarding how the money borrowed will be used.

(3) Lenders can commit to an optional portion of the total amount of a loan request.

(4) When the commitments for the loan request covers the requested amount, the request is converted to a loan. If not enough commitments can be reached, the loan request is cancelled. A borrower can have more than one request, and more than one loan at a time, but can at most make one request per day.

(5) The loan is paid through an "intermediary", typically a local department of a charity, who has a name and an address.

(6) The borrower chooses when he or she will make a payment. Every payment must be registered in the database with an amount and a date (at most one payment per loan per day). The lenders share the repayment based on how large a part of the loan they are responsible for.

(7) If the loan is not repaid before the agreed upon deadline, a new date is agreed. The database must not delete the old deadline, but save the history (the deadline can be overridden multiple times).

(8) Each lender can for each burrower save a "trust", which is a number between 0 and 100 that determines the lender's evaluation of the risk of lending money to that person. The number must only be saved for the borrowers, for whom there has been made such an evaluation.

Please draw an E-R diagram based on the business rules described above. If you make any assumptions about data that doesn't show from the problem, they must be described.

3. The following code used by an information system in a local school is written in a dynamic object-oriented language. You can see that this code suffers from repetition of the same lines in multiple places. They need your help to refactor and improve this code using OOP, without breaking any other code that uses it! (20%)

```
class Student
  attr_accessor :name
  attr_reader :must_teach

  def initialize
    @must_teach = false
  end

  def notify_enroll(semester)
    puts("#{@name} must enroll for classes in #{semester}")
  end
end

class Lecturer
  attr_accessor :name
  attr_reader :must_teach, :must_research

  def initialize
    @must_teach = true
    @must_research = false
  end

  def notify_grades(semester)
    puts("#{@name} must submit grades for #{semester}")
  end
end

class Professor
  attr_accessor :name
  attr_reader :must_teach, :must_research

  def initialize
    @must_teach = true
    @must_research = true
  end

  def notify_grades(semester)
    puts("#{@name} must submit grades for #{semester}")
  end

  def notify_grant(semester)
    puts("#{@name} must submit a grant in #{semester}")
  end
end
```

**You must rewrite this code as follows:**

- Your new code must work perfectly with the existing test code that uses it (see the test code on the next page).

- Take advantage of object-orientated techniques that allow you to abstract away 國立

repetition in your code (see sample OOP code on the next page).

- As far as you can, try to follow the syntax of the same programming language

Here is working test code that must run even with your refactored code.

```
# TEST CODE
lee = Student.new
lee.name = 'Li-Hsieh Huang'
hsu = Lecturer.new
hsu.name = 'Powei Hsu'
lu = Professor.new
lu.name = 'Peishan Lu'
everyone = [lee, hsu, lu]

for person in everyone
  if (not person.must_teach)
    person.notify_enroll('Spring 2015')
  end

  if (person.must_teach)
    person.notify_grades('Fall 2014')
  end

  if (person.must_teach) && (person.must_research)
    person.notify_grant('Spring 2015')
  end
end
# => Li-Hsieh Huang must enroll for classes in Spring 2015
# => Powei Hsu must submit grades for Fall 2014
# => Peishan Lu must submit grades for Fall 2014
# => Peishan Lu must submit a grant in Spring 2015
```

To help you understand the syntax, here is sample OOP code that uses inheritance:

```
# SAMPLE OOP CODE:
class Automobile
  attr_accessor :brand          # define a public instance variable
  attr_reader :travel           # define a private instance variable

  def initialize()              # define constructor (initialize)
    @travel = 'ground'          # set an instance variable
  end
end

class Car < Automobile          # Car inherits from Automobile class
  attr_accessor :model
  attr_reader :num_wheels

  def initialize()
    super()                     # Calls the parent' constructor method
    @num_wheels = 4
  end

  def describe()
    puts "The #{@brand} #{@model} has #{@num_wheels} wheels"
  end
end

my_car = Car.new()
my_car.brand = 'Luxgen'
my_car.model = 'U6 Turbo'
my_car.describe()

# => "The Luxgen U6 Turbo has 4 wheels"
```