

類 科：資訊處理
科 目：程式設計概要
考試時間：1 小時 30 分

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

一、請使用 C、C++、C#、Java 或 Python 程式語言，撰寫一個進位制轉換程式的方法，方法接受一個合法的整數，並以參數形式回傳十進位制分別轉成以 string 資料型別表示的二、八、及十六進位制結果；主程式負責可以持續接受輸入，檢查輸入為合法的整數值後，呼叫轉換程式的方法，並顯示結果，如果輸入不是合法的整數，結束程式的執行。執行範例如下：(25 分)

測試一：輸入31 → 呼叫tenBaseTo2_8_16base(31)，回傳 11111, 37, 1F

測試二：輸入168 → 呼叫tenBaseTo2_8_16base(127)，回傳 10101000, 250, A8

限制一：所撰寫的方法內，不可以使用任何程式語言內建或第三方提供的套件，即Python/Java不可以使用import，C++、C#不可以使用using再加入套件……

限制二：不可使用recursive的方式，亦不可以使用goto

二、如下的 Python 程式，請說明執行後，如果輸入"guava"顯示的結果為何？又如果輸入"kiwi"顯示的結果又為何？如無法正確執行，請說明原因，及應如何修正？(25 分)

假設 createLL方法會正確地建立單向連結的linked list。

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
def linearSearchR(node, target, index=0, callCount=0):
    callCount += 1 # increment the call count
    if node is None: return (False, -1, callCount)
    if node.data == target: return (True, index, callCount)
    return linearSearchR(node.next, target, index+1, callCount)
def linearSearchL(node, target):
    current = head
    index = 0
    comparisonCount = 0
    while current is not None:
        comparisonCount += 1 # increment the comparison count
        if current.data == target: return (True, index, comparisonCount)
        current = current.next
        index += 1
    return (False, -1, comparisonCount)

if (__name__ == "__main__"):
    strings = ["apple", "banana", "cherry", "guava", "mango", "pineapple"]
    head = createLL(strings)
    target = input("Please input the searching target --> ")

    found, index, call_count = linearSearchR(head, target)
    print(f"\nNumber of recursive calls made --> {call_count}")
    if found: print(f"'{target}' found in the linked list at index {index}.")
    else: print(f"'{target}' not found in the linked list.")

    found, index, comparisonCount = linearSearchL(head, target)
    print(f"\nNumber of comparisons made --> {comparisonCount}")
    if found: print(f"'{target}' found in the linked list at index {index}.")
    else: print(f"'{target}' not found in the linked list.")
```

三、請就如下的 Java 程式，說明程式中各方法（包括建構子）執行的順序，及最後顯示的結果。註：執行程序說明，如 class LinkedList 建構子 (head, tail) → …… (30 分)

```
class Node {
    int value;
    Node next;
    public Node(int vv, Node nn) {
        this.value = vv;
        this.next = nn;
        System.out.println("class Node \tconstructor \tvalue " + vv); }
}

class LinkedList {
    protected Node head;
    protected Node tail;
    public LinkedList() {
        this.head = null;
        this.tail = null;
        System.out.println("class LL \tconstructor()"); }
    public LinkedList(Node head, Node tail) {
        this.head = head;
        this.tail = tail;
        System.out.println("class LL \tconstructor(head, tail)"); }
    public void addToHead(Node newNode) {
        if (head == null) { head = newNode; tail = newNode; }
        else { newNode.next = head; head = newNode; }
        System.out.println("class LL \tmethod \t\taddtoHead " + newNode.value); }
    public void addToTail(Node newNode) {
        if (tail == null) { head = newNode; tail = newNode; }
        else { tail.next = newNode; tail = newNode; }
        System.out.println("class Node LL \tmethod \t\taddtoTail " + newNode.value); }
}

class SinglyLinkedList extends LinkedList {
    public SinglyLinkedList(Node head, Node tail) {
        super(head, tail);
        System.out.println("class SLL \tconstructor(head, tail)"); }
}
```

```
public class PQ3_LL {
    public static void main(String[] args) {
        int [] data = {90, 50, 70, 30, 40};
        SinglyLinkedList sllObj = new SinglyLinkedList(null, null);
        for (int i=0; i<data.length; i++)
        {   if (i%2==0) { sllObj.addToHead(new Node(data[i], null)); }
            else       { sllObj.addToTail(new Node(data[i], null)); }
        }
    }
} // End of main class
```

OO test program

四、請撰寫 HTML 程式，執行後可以呈現如下的頁面，其中"Year"是最多四位數的整數值，Submit 按鈕後，以 POST 形式傳送表單的資料到後端 "addMovie.php"處理。(20 分)

Add Movie

Name of Movie:

Year:

Movie Genre: Action ▼

Rating:

☐ Kids

☐ PG13

☒ Adult

Action

Comedy

Drama

Horror

Fantasy

Romance

Science Fiction