

高點

堅持夢想 全力相挺

公職 快速通關 EXPRESS >>>

Pass!

高普考准考證 就是你的 **VIP券**

弱科健檢 

加入【高點·高上生活圈】可免費預約參加 ▶▶▶



113/7/5-14 優惠再升級！

【面授/網院】全修課程最高折 **5,000** 元，再提供線上補課
考取班享專案優惠價，最高折 **10,000** 元
分眾課另享現金折扣

【雲端函授】全修課程最高折 **3,000** 元

113/7/31 前 **商管** **會計** **資訊** **地政** 享准考證優惠！

113 地方特考 衝刺

【總複習】網院：特價 2,000 元起、雲端：特價 3,000 元起

【申論寫作正解班】網院：特價 3,000 元起/科、雲端：特價 6 折起/科

【經典題庫班】網院：特價 2,500 元起/科、雲端：特價 6 折起/科

114 高普考 達陣

【全修課程】面授/網院：高考特價 46,000 元起、普考特價 41,000 元起
雲端：高考特價 51,000 元、普考特價 46,000 元

【考取班】高考：特價 75,000 元、普考：特價 65,000 元 (限面授/網院)

【狂作題班】面授：特價 6,000 元/科

單科 加強方案

【113年度】網院：定價 5 折起、雲端：定價 7 折起

【114年度】面授/網院：定價 65 折起、雲端：定價 85 折

※優惠詳情依各分班櫃檯公告為準

《程式設計概要》

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

| | |
|-------------|---|
| 試題評析 | 本試題主要的重點是迴圈，不能使用遞迴。十進位轉二進位、八進位與十六進位採用連除法，因為答案是在連續除法後先得到餘數後顯示，因此使用有字串物件的語言相對容易處理，因此本題採用Java語言來回答。本題難度不高，用功的學員可得高分。 |
| 考點命中 | 《高點·高上程式設計概要講義》第一回，許振明編撰，頁80、170。 |

答：

```
public class Test{
    public static void tenBaseTo2_8_16base(int n){
        int n2,n8,n16;
        String str="";
        if(n<0)    System.out.println("輸入錯誤\n");
        else{
            n2=n;
            while(n2>0){
                str = (n2%2) + str;
                n2/=2;
            }
            System.out.print(str + ", ");
            str = "";
            n8=n;
            while(n8>0){
                str = (n8%8) + str;
                n8/=8;
            }
            System.out.print(str + ", ");
            str = "";
            n16=n;
            while(n16>0){
                switch(n16%16){
                    case 10: str = "A" + str;
                            break;
                    case 11: str = "B" + str;
                            break;
                    case 12: str = "C" + str;
                            break;
                    case 13: str = "D" + str;
```

```

        break;
    case 14: str = "E" + str;
        break;
    case 15: str = "F" + str;
        break;
    default:
        str = (n16%16) + str;
    }
    n16/=16;
}
System.out.print(str);
}
}

public static void main(String [] arg){

    tenBaseTo2_8_16base(168);
}
}

```

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", "guave", "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.")

```

試題評析

本試題是Python版本的線性搜尋法(linear search)，採用迴圈與遞迴兩個函數製作，測試時使用兩個測試案例，一個可以搜尋到資料(guava)，另一個是搜尋不到資料(kiwi)，兩個測試案例均可正常執行並得到結果，仔細作答可獲取高分。

考點命中

《高點·高上程式設計概要講義》第二回，許振明編撰，頁231-234。

答：

(一)

Please input the searching target --> guava

Number of recursive calls made --> 4

'guava' found in the linked list at index 3.

Number of comparisons made --> 4

'guava' found in the linked list at index 3.

(二)

Please input the searching target --> kiwi

Number of recursive calls made --> 7

'kiwi' not found in the linked list.

Number of comparisons made --> 6

'kiwi' not found in the linked list.

註：完整程式如下

class Node:

def __init__(self, data) -> None:

self.data = data

self.next = None

def linearSearchR(node, target, index=0, callCount=0):

callCount += 1

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

if current.data == target:

return (True, index, comparisonCount)

current = current.next

index += 1

return (False, -1, comparisonCount)

def createLL(str):

myhead = None

for fruit in str:

this_node = Node(fruit)

if myhead is None:

myhead = this_node

else:

ptr = myhead

while ptr.next is not None:

ptr = ptr.next

【版權所有，重製必究！】

```

        ptr.next = this_node
    return myhead

```

```

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 P03_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

```

| | |
|------|--|
| 試題評析 | 本試題使用Java語言製作單向鏈結串列(Singly Linked List)，並提供兩個函數，將節點由鏈結串列前端加入addToHead()與將節點由鏈結串列後端加入addToTail()。主程式由整數陣列讀取資料後，透過上述的兩個函數建立單向鏈結串列，其中利用繼承與建構子的設計方式將整數陣列轉換成鏈結串列，只要對物件導向的概念清楚就可以獲取高分。 |
| 考點命中 | 《高點·高上程式設計概要講義》第二回，許振明編撰，頁165-168。 |

答：

```

class LL      constructor(head, tail)
class SLL    constructor(head, tail)
class Node   constructor      value 90
class LL     method          addToHead 90
class Node   constructor      value 50
class LL     method          addToTail 50
class Node   constructor      value 70
class LL     method          addToHead 70
class Node   constructor      value 30
class LL     method          addToTail 30
class Node   constructor      value 40
class LL     method          addToHead 40

```

說明：

```

public class PQ3_LL{
    public static void main(String [] args){
        int [] data = {90, 50, 70, 30, 40};
        SinglyLinkedList sllObj = new SinglyLinkedList(null, null);
        // 呼叫SinglyLinkedList(Node head, Node tail) -> 呼叫 LinkedList(Node head, Node tail)
        for(int i=0; i<data.length; i++){
            if(i%2==0){
                sllObj.addToHead(new Node(data[i], null));
                // 分別執行 90, 70, 40 的建構子Node(int vv, Node nn)後，再呼叫addToHead(Node newNode)函數
            }else{
                sllObj.addToTail(new Node(data[i], null));
                // 分別執行 50, 30 的建構子Node(int vv, Node nn)後，再呼叫addToTail(Node newNode)函數
            }
        }
    }
}

```

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)

【版權所有，重製必究！】

| | |
|------|---|
| 試題評析 | 本試題主要測試對於HTML表單(form)的用法以及使用post方式將表單資料送出給指定的後端程式addMovie.php，這是一個web程式的基礎試題，對於表單中常見的物件，如：文字方塊(text)、查核方塊(checkbox)、選鈕(radio button)等，學員應該要熟悉這些基本用法，本試題一般學員可輕鬆獲取高分。 |
| 考點命中 | 1. 《高點·高上程式設計概要講義》第二回，許振明編撰，頁7-8。 2. 《高點·高上113年程式設計概要模擬考》第四題。 |

答：

```

<html>
<body>
Add Movie<br>
<form method="post" action="addMovie.php">
Name of Movie: <input type="text" name="movie_name"><br>
Year: <input type="number" id="year" name="year" value="2024" min="2020" max="2030"><br>
Movie Genre: <select id="genre" name="genre">
  <option value="Action">Action</option>
  <option value="Comedy">Comedy</option>
  <option value="Drama">Drama</option>
  <option value="Horror">Horror</option>
  <option value="Fantancy">Fantancy</option>
  <option value="Romance">Romance</option>
  <option value="ScienceFiction">Science Fiction</option>
</select><br>
Rating:<br>
<input type="radio" name="rating" value="Kids">Kids<br>
<input type="radio" name="rating" value="PG13">PG13<br>
<input type="radio" name="rating" value="Adult" checked>Adult<br><br>
<input type="submit" value="submit">
</form>
</body>
</html>

```

【版權所有，重製必究！】

高
點

高點資訊公職書系 上榜者搶分推薦！

重點整理書系—萃取考試重點、綜合模擬題&整合觀念混淆題。

解題書系列—收錄高頻率試題、實力養成題庫，短時間掌握命題脈絡。

重點整理

| 書名 | 作者 | 定價 |
|--------------|---------|-----|
| 國文(測驗)國考必勝秘笈 | 尹宸 | 620 |
| 國文(作文)國考必勝秘笈 | 尹宸 | 550 |
| 中華民國憲法 | 歐律師 | 680 |
| 憲法 | 王肇基 | 650 |
| 法學緒論 | 徐英智 | 680 |
| 資料結構 | 王致強 | 680 |
| 資料庫應用 | 向宏 | 700 |
| 計算機概要 | 余強 | 680 |
| 資料處理(概要) | 柯霖廷、許溥祐 | 580 |



解題完全制霸

| 書名 | 作者 | 定價 |
|-----------------------|-----|-----|
| 國文(作文/測驗)解題攻略 | 簡正崇 | 580 |
| 國文/測驗解題一本通 | 楊昕 | 650 |
| 英文解題完全制霸 | 林惠華 | 600 |
| 法學緒論好好考 | 嶺律師 | 450 |
| 憲法測驗題好好考 | 嶺律師 | 580 |
| 程式設計概要歷屆試題精解 | 向宏 | 500 |
| 資通網路與安全(概要) 歷屆試題詳解 | 張又中 | 預版中 |
| 資訊管理歷屆試題詳解 | 蕭維文 | 預版中 |

※定價以版權頁為準！



※最新考情及考試科目以考選部公告為準！

※線上試讀請至高點網路書店，第一次加入會員還可享 \$50 購書贊助金！

高點文化事業
publish.get.com.tw



113/7/11-8/31年中慶特惠中
手刀購買，快至高點網路書店