

## 考試規定

- 請勿作弊，作弊者 0 分
- 請將所有題目撰寫在同一檔案中，並以 學號.c 的方式命名
- 請在各題目作答程式碼開頭，註解該題題號
- 題目共有三大題，第一大題有四個小題，單一題目的分數僅有全拿與零分兩種
- 請按照題目給定的格式輸出，若未依格式輸出者一律扣十分

## 考試題目

Q1.

請建立一個個人資料管理程式,並包含以下功能

- A. 新增一筆資料 (20%)
- B. 刪除一筆資料 (20%)
- C. 搜尋特定資料 (20%)
- D. 可多次選擇功能或是選擇結束程式 (10%)

而在資料格式方面,請使用以下 `structure` 進行實作

```
struct PersonInfo
{
    int ID; // ID
    char* name; // 姓名
    int age; // 年紀
    int gender; // 性別(1=男,2=女)
    int birthday[3]; //出生日期,birthday[0]代表年、birthday[1]代表月、birthday[2]代表日
    personInfo* LeftChild; // 在 binary search tree 中的左方子節點
    personInfo* RightChild; // 在 binary search tree 中的右方子節點
}
```

功能方面說明如下

- A. 新增一筆資料

程式需要從文字檔中讀取資料,並存放至一個 `binary search tree` 中。此時需要讓用戶輸入資料內容,並將新增的資料增加至 `binary search tree` 中。並將資料 寫入檔案中。

B. 刪除一筆資料

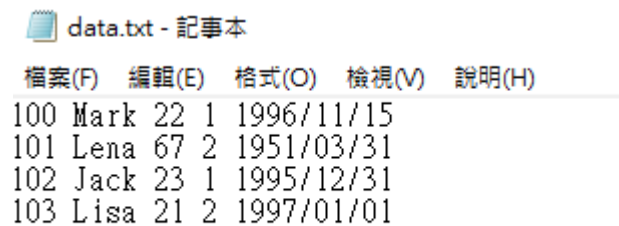
程式需要從文字檔中讀取資料,並存放至一個 **binary search tree** 中。此時需要讓用戶輸入想要刪除的資料的 **ID**,並將該筆資料刪除,最後將新的資料表寫入檔案。

C. 搜尋特定資料

程式需要從文字檔中讀取資料,並存放置一個 **array** 中。此時需要讓用戶輸入想要搜尋的欄位,並輸入該欄位的值。如有找到,則列出對應到的資料內容,若否,則輸出 "Not found\n"。

注意,本程式須讓使用者多次選擇功能並進行操作,且要讓使用者在選擇功能時亦可選擇跳出程式。

輸出檔案內容表示如下(由左至右為 **ID**、姓名、年紀、性別、出生日期)



```
data.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
100 Mark 22 1 1996/11/15
101 Lena 67 2 1951/03/31
102 Jack 23 1 1995/12/31
103 Lisa 21 2 1997/01/01
```

## 輸出範例

```
1. Add
2. Delete
3. Search
4. Exit
Option: 1
Please input new data
ID: 100
Name: Mark
Age: 22
Gender: 1
Birthday: 1996 11 15
Data created
1. Add
2. Delete
3. Search
4. Exit
Option: 2
Please input ID
ID: 103
Data deleted
1. Add
2. Delete
3. Search
4. Exit
Option: 3
Please input field to search (1.ID 2.Name 3.Age 4.Gender 5.Birthday): 2
Please input name: Mark
ID: 100
Name: Mark
Age: 22
Gender: 1
Birthday: 1996 11 15
Search completed
1. Add
2. Delete
3. Search
4. Exit
Option: 4
Have a good day
```

Q2.

輸入兩個已排序的 linked list，產生一個 linked list 包含從頭到尾最大的求和路徑。  
產生的 linked list 可能包含兩個輸入的 linked list 的節點。

當我們要產生新的 linked list 時，我們只會在兩個 linked list 的交點做切換(交點指的是兩個 linked list 裡的相同數字)，如果兩個 linked list 兩個交點之間個別的和一樣的話不用切換，如 1->10->15->50 和 1->25->50，本題需要使用 linked list 撰寫，用其他方式撰寫不給分。(25%)

### Input:

List1 = 1->3->30->90->120->240->511

List2 = 0->3->12->32->90->125->240->249

需要輸入的是紅色的部分(1->3->30->90->120->240->511)

### Output:

list = 1->3->12->32->90->125->240->511

在 3 和 240 時切換得到上列最大的求和路徑

輸入輸出範例

```
List1 = 1->3->30->90->120->240->511
List2 = 0->3->12->32->90->125->240->249
Maximum sum linked list = 1->3->12->32->90->125->240->511
```

Q3.

在本題中，題目會先給你一個包含小括號()及中括號[]的字串。當字串符合下列條件時我們稱他為正確的運算式：(25%)

1.該字串為一個空字串

2.如果 A 和 B 都為正確的運算式，則 A B 也為正確的運算式，

3.如果 A 為正確的運算式，則(A)及[A]都為正確的運算式。

現在，請你寫一支程式可以讀入這類字串並檢查它們是否為正確的運算式。字串的最大長度為 128 個字元。

### Input:

輸入的第一列為正整數 n，代表接下來有 n 列待測資料。

## Output:

檢查每列待測資料，如果正確輸出 Yes，否則輸出 No。

輸入範例

3

([])

(([])))

([()])()

第一個數字 3 代表接下來會輸入三次括弧，接著連續輸入三串括弧，輸入完後顯示結果。

輸出範例

Yes

No

Yes

輸入輸出範例

```
3
([])
(([])))
([()])()
Yes
No
Yes
```