

Data Structure Assignment

姓名：吳威廷

學號：B10702239

一、系統開發需求

1. 使用開發環境：

CLion

2. 使用語言：

C 語言

3. 使用作業系統：

win10

4. 題目要求：

輸入檔案數量和檔案名稱並建立 linked

list 和一個 listAry 把所有 list 給連起來把這個結構做

完之後再做一些與這個結構相關的 function。

二、程式功能說明

1. Main

```
1  #include <stdio.h>
2      #include <stdlib.h>
3  #include "string.h"
4
5      #define MAX_SIZE 50
6
7  typedef struct listNode {
8      int value;
9      struct listNode *next;
10 } node;
11
196 int main() {
197     printf( _Format: "\n'c' to print how many numbers in each list\n");
198     printf( _Format: "'p' to print all of the lists\n");
199     printf( _Format: "'l' to find the largest number in all lists\n");
200     printf( _Format: "'d' to delete the duplicated number in each list\n");
201     printf( _Format: "'n' to create the new lists\n");
202     printf( _Format: "'0' to exit this program\n");
203     createArrayList();
204 }
```

Main 先說明了使用者按下甚麼鍵之後程式會做出麼樣的動作。

當按下 p 時會 print 出所有的 list。

當按下 l(英文字母)時會找出所有 list 裡面最大的數值。

當按下 d 時會把每一條 linked list 裡面的重複數字給移除。

當按下 n 時會把之前的 list 給移除且重新輸入檔案數字和檔案

名稱以建立下一個新的結構。

當按下 0(數字)時會結束此程式

2. Create List Function

```
129 void createArrayList() {
130     int listNumber = 0, number = 0, count = 0, sign = 1;
131     char str;
132     node *temp, *tail, *list = NULL, *listArray = NULL;
133     FILE *file;
134
135     printf(_Format: "\nIn Create List Function\n");
136     printf(_Format: "Please input the number of files :");
137     scanf(_Format: "%d", &listNumber);
138     char **fileName = (char **) malloc(_Size: sizeof(char *) * listNumber);
139
140     for (int i = 0; i < listNumber; ++i) fileName[i] = (char *) malloc(_Size: sizeof(char) * MAX_SIZE);
141
142     for (int i = 0; i < listNumber; ++i) {
143         printf(_Format: "Please input the name of No.%d file :", i + 1);
144         scanf(_Format: " %s", fileName[i]);
145     }
146
147     listArray = (node *) malloc(_Size: sizeof(node) * listNumber);
148
149     for (int k = 0; k < listNumber; k++) { // Create List
150         strcat(fileName[k], _Source: ".txt");
151
152         if (!(file = fopen(fileName[k], _Mode: "r"))) {
153             printf(_Format: "Fail to open this file\n");
154             exit(EXIT_FAILURE);
155         }
156
157         temp = (node *) malloc(sizeof(node));
158         tail = temp;
159
160         while ((str = (char) fgetc(file)) != EOF) {
161
162             if (str == ' ') {
163                 list = (node *) malloc(sizeof(node));
164                 temp->next = list;
165                 list->value = number * sign;
166                 temp = temp->next;
167                 count++;
168                 number = 0;
169                 sign = 1;
170             } else if (str >= '0' && str <= '9') {
171                 number = 10 * number + (str - '0');
172             } else if (str == '-') {
173                 sign = -1;
174             } else {
175                 number = 0;
176             }
177         }
178
179         listArray[k].value = ++count;
180         listArray[k].next = (count < 2) ? tail : tail->next;
181         temp->next = tail; //deal with the last number of list
182         tail->value = number * sign;
183         tail->next = NULL;
184         number = 0;
185         count = 0;
186
187         sign = 1;
188         fclose(file);
189     }
190
191     for (int i = 0; i < listNumber; ++i) free(fileName[i]);
192     free(fileName);
193     interface(listArray, listNumber, &createArrayList);
194 }
```

使用者輸入要建立 list 的檔案數量和檔案名稱，之後程式開始開檔案並且把資料建好已完成系統開發需求所要的資料結構形式。

3. Print List Function

```
69
70 void printList(node *listArray, int listNumber) {
71     node *list = NULL;
72
73     printf(_Format: "\nIn print Function\n");
74
75     for (int k = 0; k < listNumber; ++k) {
76         list = listArray[k].next;
77         printf(_Format: "No.%d List :", k + 1);
78
79         for (int i = 0; i < listArray[k].value; ++i) {
80             printf(_Format: "%d ", list->value);
81             list = list->next;
82         }
83
84         printf(_Format: "\n");
85     }
86
87     printf(_Format: "\n");
88 }
```

此 function 用來印出已經建好的所有 list。

4. Print Count Function

```
59
60 void printCount(node *listArray, int listNumber) {
61     printf(_Format: "\nIn Print Count Function\n");
62
63     for (int i = 0; i < listNumber; ++i) {
64         printf(_Format: "Numbers of No.%d list : %d\n", i + 1, listArray[i].value);
65     }
66
67     printf(_Format: "\n");
68 }
```

此 function 用來 print 出每個 list 有多少數字。

5. Print Largest Function

```
43 int largestNumber(node *listArray) {
44     int maxNumber = INT_MIN;
45     node *list = NULL;
46
47     for (int i = 0; i < 4; ++i) {
48         list = listArray[i].next;
49
50         for (int j = 0; j < listArray[i].value; ++j) {
51             maxNumber = (maxNumber > list->value) ? maxNumber : list->value;
52             list = list->next;
53         }
54     }
55
56     return maxNumber;
57 }
58 }
```

此 function 用來 print 出所有 list 裡面最大的數字。

6. Delete Duplicated Numbers

```
11
12 node *deleteDuplicated(node *listArray, int listNumber) {
13     node *prePtr, *curPtr, *data;
14     int deleteNumber = 0;
15
16     printf(_Format: "\nDeleting duplicated.....\n");
17
18     for (int i = 0; i < listNumber; i++) {
19
20         for (data = listArray[i].next; data != NULL; data = data->next) {
21
22             for (prePtr = data, curPtr = data->next; curPtr != NULL; prePtr = curPtr, curPtr = curPtr->next) {
23
24                 if (curPtr->value == data->value) {
25                     prePtr->next = curPtr->next;
26                     free(curPtr);
27                     curPtr = prePtr;
28                     deleteNumber++;
29                 }
30             }
31         }
32     }
33
34     listArray[i].value -= deleteNumber;
35     deleteNumber = 0;
36 }
37
38
39 printf(_Format: "Finished !!!!\n\n");
40 return listArray;
41 }
```

此 function 用來刪除每條 list 裡面重複的數值並更新存在 listArray 裡面的 Count 數值。

7. Interface Function

```
90 void interface(node *listArray, int listNumber, void (*ptr)()) {
91     char str;
92
93     while (1) {
94         printf(_Format: "Key the command to do specific things : ");
95         scanf(_Format: " %c", &str);
96
97         switch (str) {
98             case 'c':
99                 printCount(listArray, listNumber);
100                 break;
101             case 'p':
102                 printList(listArray, listNumber);
103                 break;
104             case 'l':
105                 printf(_Format: "\nThe largest number is %d\n", largestNumber(listArray));
106                 break;
107             case 'd':
108                 deleteDuplicated(listArray, listNumber);
109                 break;
110             case 'n':
111                 for (int i = 0; i < listNumber; ++i) free(listArray[i].next);
112                 free(listArray);
113                 (*ptr)();
114                 break;
115             case '0':
116                 printf(_Format: "\nProgram finished !");
117                 for (int i = 0; i < listNumber; ++i) free(listArray[i].next);
118                 free(listArray);
119                 exit(EXIT_SUCCESS);
120             default :
121                 printf(_Format: "Cannot find the command\n");
122                 break;
123         }
124     }
125 }
126
127 }
128
```

此 function 用來判斷使用者輸入的指令是什麼，根據上述的功能來做應對的操作。

三、測試結果

```
'c' to print how many numbers in each list
'p' to print all of the lists
'l' to find the largest number in all lists
'd' to delete the duplicated number in each list
'n' to create the new lists
'0' to exit this program
```

In Create List Function

```
Please input the number of files :2
Please input the name of No.1 file :a
Please input the name of No.2 file :b
Key the command to do specific things : p
```

In Print Function

```
No.1 List :-3
No.2 List :2 2 -6 8 8 6 10 -13 15 17 17 -17
```

Key the command to do specific things : c

In Print Count Function

```
Numbers of No.1 list : 1
Numbers of No.2 list : 12
```

Key the command to do specific things : l

The largest number is 17

Key the command to do specific things : d

```
Deleting duplicated.....
Finished !!!!
```

Key the command to do specific things : p

In Print Function

```
No.1 List :-3
No.2 List :2 -6 8 6 10 -13 15 17 -17
```

Key the command to do specific things : c

In Print Count Function

```
Numbers of No.1 list : 1
Numbers of No.2 list : 9
```

Key the command to do specific things : n

In Create List Function

```
Please input the number of files :2
Please input the name of No.1 file :c
Please input the name of No.2 file :d
Key the command to do specific things : p
```

In Print Function

```
No.1 List :2 1 3 5 7 9 1 3 5 7 9
No.2 List :10 20 30 40 11 111 111 11 1111 111 1111
```

Key the command to do specific things : c

In Print Count Function

```
Numbers of No.1 list : 11
Numbers of No.2 list : 11
```

Key the command to do specific things : l

The largest number is 1111

Key the command to do specific things : d

```
Deleting duplicated.....
Finished !!!!
```


Key the command to do specific things : p

In Print Function

No.1 List :2 1 3 5 7 9

No.2 List :10 20 30 40 11 111 1111

Key the command to do specific things : c

In Print Count Function

Numbers of No.1 list : 6

Numbers of No.2 list : 7

Key the command to do specific things : d

Deleting duplicated.....

Finished !!!!

Key the command to do specific things : p

In Print Function

No.1 List :2 1 3 5 7 9

No.2 List :10 20 30 40 11 111 1111

Key the command to do specific things : c

In Print Count Function

Numbers of No.1 list : 6

Numbers of No.2 list : 7

Key the command to do specific things : l

The largest number is 1111

Key the command to do specific things : 0

Program finished !

四、效能分析

1. Create List Function

此 function 的時間複雜度為每個 list 裡面的字串數量相加，因為每個 list 裡面的數字不固定所以 while 迴圈在跑的時候會到 file 的最尾端，根據 file 數量的不同和各個檔案的數字的不同，此 function 的時間複雜度會不同。

2. Print List Function

此 function 的時間複雜度為每個 list 的數字個數相加。

3. Print Count Function

此 function 的時間複雜度為檔案次數個，因為把每個 list 的個數 print 出來只需要把 listAry 裡面的值 print 出來即可，listAry 的數量正是檔案數量。

4. Print Largest Function

此 function 的時間複雜度為每個 list 的數字個數和，因為要找到所有 list 裡面的數字必須把每個 list 裡面的數字給掃過一次才能找到最大值，所以此 function 的時間複雜度為所有 list 的數字個數和。

5. Delete Duplicated Numbers

此 function 的時間複雜度為每一條 list 的數字個數平方相加，因為刪除重複數字的演算法時間複雜度為 $O(n^2)$ ，所以 for 迴圈跑得次數就是每一條 list 裡面的數字個數平方在把所有的個數平方加起來就是此 function 的時間複雜度。

五、心得

經過這次的作業感覺對原本沒有很熟的 linked list 給弄熟了，在寫這個程式時也用了許多的 pointer，上學期修程式課的時候感覺還沒有對 pointer 的操作到非常熟，但是經過這次的作業讓我對 pointer 的操作更加的了解，也上網查了很多關於 pointer 的操作，相信之後的資料結構也會運用到非常多 pointer 的操作，所以一定要把這塊給弄懂、弄熟來，期望之後能在程式方面能更進步。