國立屏東大學 資訊工程系 程式設計(二)

Turnin作業4

- Turnin Code: c.hw4
- Due Date: 4/23 23:59:00 **Hard Deadline**

繳交方式說明

本次Turnin作業包含多個程式題，建議同學可以為這次作業先建立一個資料夾hw4，然後在該資料夾內再為每一題建立一個子資料夾，用以進行每一題的作答以及上傳。每一題的子資料夾名稱已寫於題目前方，請務必依照題目的規定建立子資料夾，例如第1題為p1，第2題為p2，餘依此類推。當我們完成某一個題目的作答後，就可以使用turnin指令將該題的答案上傳。以第1題為例，當我們在p1子資料夾裡完成作答後，就可以回到hw4資料夾，使用以下指令將其上傳：

[user@ws hw4]$ turnin␣c.hw4␣p1↵

當然，你也可以等到所有題目都完成後，再回到hw4資料夾，使用以下指令將所有題目都加以上傳：

[user@ws hw4]$ turnin␣c.hw4␣.↵

本文使用 ␣ 及 ↵ 代表空白字元與Enter換行字元，並且將使用者輸入的部份使用灰階方式顯示。另外，題目的執行結果中，如果出現(、)、:、;、.與,等符號，皆為英文半形！

當你完成此次作業的繳交後，可以使用turnin指令的-ls參數，查看已繳交的結果。若已經正確地依要求繳交此次作業，那麼你將可以看到類似以下的查詢結果：

```
[user@ws hw4]$ turnin -ls c.hw4
.:
total 20
drwxrwx---. 2 turninman 1669401585 4096 Mar  5 20:57 p1
drwxrwx---. 2 turninman 1669401585 4096 Mar  5 20:57 p2
drwxrwx---. 2 turninman 1669401585 4096 Mar  5 20:57 p3
drwxrwx---. 2 turninman 1669401585 4096 Mar  5 20:57 p4
drwxrwx---. 2 turninman 1669401585 4096 Mar  5 20:57 p5
drwxrwx---. 2 turninman 1669401585 4096 Mar  5 20:57 p6

./p1:
total 8
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 ReverseString.c
```

```
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 ReverseString.h

./p2:
total 8
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Sudoku.c
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Sudoku.h

./p3:
total 8
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Contact.c
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Contact.h

./p4:
total 8
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 ReplaceSubstring.c
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 ReplaceSubstring.h

./p5:
total 8
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Main.c
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Makefile
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Gomoku.c
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Gomoku.h

./p6:
total 8
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Main.c
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Makefile
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Gomoku.c
-rw-rw----. 1 turninman 1669401585 4 Mar  5 20:57 Gomoku.h
[user@ws hw4]$
```

【注意：以上的執行結果僅供參考，包含檔案上傳的日期、時間、大小等皆會依實際情況有所不同，請自行仔細檢查是否有正確繳交。】

若是發現自己繳交錯誤的同學，也可以使用以下的指令，將此次作業所有已上傳的檔案與資料夾全部清空：

[user@ws hw4]$ `turnin▲-rm▲c.hw4↵`

---

p1 反轉字串

請參考如下 Main.c 的程式碼:

```
#include <stdio.h>

#include "ReverseString.h"
```

```
int main() {
    char str[1025];

    printf("Enter a string: ");

    GetString(str);
    ReverseString(str);

    printf("Reversed string: %s\n", str);

    return 0;
}
```

同學需完成並繳交 **ReverseString.c** 與 **ReverseString.h** 的 C 語言程式，其中分別包含 ReverseString() 函式的 實作(Implementation)與其 原型(Prototype)宣告。

本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c ReverseString.o
    gcc Main.c ReverseString.o

ReverseString.o: ReverseString.c ReverseString.h
    gcc -c ReverseString.c

clean:
    rm -rf *.o *.out *.exe *~
```

本題可參考以下的執行結果：

```
[3:23 user@ws hw] ./a.out↵
Enter▲a▲string:▲Hello▲World!↵
Reversed▲string:▲!dlroW▲olleH↵
[3:23 user@ws hw] ./a.out↵
Enter▲a▲string:▲▲▲ZXCZXC▲▲▲▲▲▲▲↵
Reversed▲string:▲CXZCXZ↵
[3:23 user@ws hw] ./a.out↵
Enter▲a▲string:▲▲▲▲▲▲abcabc▲▲▲▲↵
Reversed▲string:▲cbacba↵
[3:23 user@ws hw]
```

1. 本題應繳交 ReverseString.c 與 ReverseString.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。
2. 本題輸出需剔除 **段落前後多餘空白**

3. 本題測試輸入長度 $0 \lt$ **字串長度** ≤ 1024 個字元。

p2 數獨檢查

數獨（すうどく□Sudoku□是一種數字邏輯遊戲，其名稱的意義是「每一格只有一個數字」。玩家需要將數字 1~9 填入 9*9 的方格中，在這 9*9 的方格中還需要分成 9 個大宮格（請參考右圖較粗的黑線劃分出井字的區塊產生的九個大宮格， 維基百科-數獨），規則如下：

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

1. 每一列（水平方向）必須包含 1~9 且不能留空也不能重複
2. 每一行（垂直方向）必須包含 1~9 且不能留空也不能重複
3. 每一個大宮格必須包含 1~9 且不能留空也不能重複

在本題中，測試檔將提供一些完整的 9*9 正確格式（每一格都是 1~9 間其中一個數字）數獨棋盤，在這些數獨棋盤中將使用空白作為每一行的數字分隔；使用換行作為每一列的分隔。請參考以下棋盤範例：

```
[3:23 user@ws hw] cat▲testfile_1.txt↵
5▲3▲4▲6▲7▲8▲9▲1▲2↵
6▲7▲2▲1▲9▲5▲3▲4▲8↵
1▲9▲8▲3▲4▲2▲5▲6▲7↵
8▲5▲9▲7▲6▲1▲4▲2▲3↵
4▲2▲6▲8▲5▲3▲7▲9▲1↵
7▲1▲3▲9▲2▲4▲8▲5▲6↵
9▲6▲1▲5▲3▲7▲2▲8▲4↵
2▲8▲7▲4▲1▲9▲6▲3▲5↵
3▲4▲5▲2▲8▲6▲1▲7▲9↵
[3:23 user@ws hw]
```

請參考下列 Main.c 的程式碼:

```
#include <stdio.h>
#include "Sudoku.h"
#define true 1
#define false 0
int main(){
    int board[9][9];

    for(int i=0; i<9; i++){
```

```
        for(int j=0; j<9; j++){
            scanf(" %d", &board[i][j]);
        }
    }

    (isValidSudoku(board) == true) ?
        printf("Valid\n") :
        printf("Invalid!\n");
}
```

請觀察主程式，同學必須完成 isValidSudoku() 函式，判斷主程式取回的數獨棋盤是否合乎上述數獨規則。同學需完成並繳交 Sudoku.c 與 Sudoku.h 的 C 語言程式，其中**分別包含** isValidSudoku() 函式的**實作**（**Implementation**）與其 **原型**（**Prototype**）宣告。

本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c Sudoku.o
    gcc Main.c Sudoku.o
Sudoku.o: Sudoku.c Sudoku.h
    gcc -c Sudoku.c
clean:
    rm -f *.o *~ a.out
```

本題可參考以下的執行結果：

```
[3:23 user@ws hw] ./a.out▲<▲testfile_1.txt↵
Valid↵
[3:23 user@ws hw] ./a.out▲<▲testfile_2.txt↵
Invalid!↵
[3:23 user@ws hw]
```

本題測試檔路徑：

- /home/stu/public/c2025s/c.hw4/p2/testfile_1.txt
- /home/stu/public/c2025s/c.hw4/p2/testfile_2.txt

本題應繳交 Sudoku.c 與 Sudoku.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。

p3 * 通訊錄

請參考如下 Main.c 的程式碼:

```
#include <stdio.h>
#include "Contact.h"

Contact contacts[MAX_CONTACTS];

int main() {
    int quit = 0;

    printf("Simple Address Book Program\n");
    Prompt_Help();
    while(!quit) {
        char choice = 0;
        printf("Option >> ");
        scanf(" %c", &choice);

        Prompt_SelectedOption(choice);
        switch(choice) {
            case 'a':
                Contact_Add(contacts);
                break;
            case 'd':
                Contact_Delete(contacts);
                break;
            case 'l':
                Contact_List(contacts);
                break;
            case 'u':
                Contact_Update(contacts);
                break;
            case 'h':
                Prompt_Help();
                break;
            case 'q':
                quit = 1;
                break;
            default:
                Prompt_Hint("Invalid option\n");
                break;
        }
    }
    printf("bye!\n");
    return 0;
}
```

同學需完成並繳交 **Contact.c** 與 **Contact.h** 的 C 語言程式，其中分別包含 **Contact_Add()**、**Contact_Delete()**、**Contact_Update()** 以及 **Contact_List()** 函式的 實作（Implementation）與其 原型（Prototype）宣告。

此外，本題另有提供以下檔案：

1. Required.h：本題必要的型態宣告

2. Prompt.h：輸出規定的格式所需要用到的函式定義
3. Prompt.c：輸出規定的格式所需要用到的函式實作

上述檔案路徑如下：

- /home/stu/public/c2025s/c.hw4/p3/Required.h
- /home/stu/public/c2025s/c.hw4/p3/Prompt.h
- /home/stu/public/c2025s/c.hw4/p3/Prompt.c

```
typedef struct {
    char year[5];
    char month[3];
    char day[3];
} Date;

typedef struct {
    char name[16];
    Date birthdate;
    char phone[11];
} Contact;

#define MAX_CONTACTS 5
```

```
#include "Required.h"

void Prompt_Help();
void Prompt_GridTitle();
void Prompt_SingleContact(Contact contact);
void Prompt_SelectedOption(char c);
void Prompt_Hint(const char* hint);
```

```
#include <stdio.h>
#include "Prompt.h"

void Prompt_Help() {
    printf("[ %-3s| %-27s]\n", "a.", "Add new contact");
    printf("[ %-3s| %-27s]\n", "d.", "Delete contact");
    printf("[ %-3s| %-27s]\n", "l.", "List all contacts");
    printf("[ %-3s| %-27s]\n", "u.", "Update contact");
    printf("[ %-3s| %-27s]\n", "h.", "Help");
    printf("[ %-3s| %-27s]\n", "q.", "Quit");
}
```

```
void Prompt_SingleContact(Contact contact) {
    printf("[ %-15s | %s-%s-%s%5s | %-15s ]\n",
        contact.name,
        contact.birthdate.year, contact.birthdate.month,
contact.birthdate.day, "",
        contact.phone
    );
}

void Prompt_GridTitle() {
    printf("[ %-15s | %-15s | %-15s ]\n", "Name", "Birthdate", "Phone");
}

void Prompt_Hint(const char* hint) {
    printf("===> Hint: %s", hint);
}

void Prompt_SelectedOption(char c) {
    printf("Selected option: (%c)\n", c);
}
```

本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c Contact.o Prompt.o
    gcc Main.c Contact.o Prompt.o

Contact.o: Contact.c Contact.h
    gcc -c Contact.c

Prompt.o: Prompt.c Prompt.h
    gcc -c Prompt.c

clean:
    rm -rf *.o *.out *.exe *~ .*
```

本題可參考以下的執行結果：

```
[3:23 user@ws hw] ./a.out↵
Simple▲Address▲Book▲Program↵
[▲a.▲|▲Add▲new▲contact▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲d.▲|▲Delete▲contact▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲l.▲|▲List▲all▲contacts▲▲▲▲▲▲▲▲▲▲]↵
[▲u.▲|▲Update▲contact▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲h.▲|▲Help▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲q.▲|▲Quit▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲]↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
```

```
===>▲Hint:▲List▲is▲empty↵
Option▲>>▲l↵
Selected▲option:▲(l)↵
===>▲Hint:▲No▲records▲were▲found↵
Option▲>>▲u↵
Selected▲option:▲(u)↵
===>▲Hint:▲List▲is▲empty↵
Option▲>>▲h↵
Selected▲option:▲(h)↵
[▲a.▲|▲Add▲new▲contact▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲d.▲|▲Delete▲contact▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲l.▲|▲List▲all▲contacts▲▲▲▲▲▲▲▲▲▲]↵
[▲u.▲|▲Update▲contact▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲h.▲|▲Help▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲]↵
[▲q.▲|▲Quit▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲]↵
Option▲>>▲f↵
Selected▲option:▲(f)↵
===>▲Hint:▲Invalid▲option↵
Option▲>>▲D↵
Selected▲option:▲(D)↵
===>▲Hint:▲Invalid▲option↵
Option▲>>▲L↵
Selected▲option:▲(L)↵
===>▲Hint:▲Invalid▲option↵
Option▲>>▲U↵
Selected▲option:▲(U)↵
===>▲Hint:▲Invalid▲option↵
Option▲>>▲H↵
Selected▲option:▲(H)↵
===>▲Hint:▲Invalid▲option↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲harper↵
Birthdate▲(YYYY▲MM▲DD):▲2024▲11▲27↵
Phone▲number:▲0887654321↵
===>▲Hint:▲Contact▲added↵
Option▲>>▲l↵
Selected▲option:▲(l)↵
Found▲1▲record▲in▲total↵
[▲Name▲▲▲▲▲▲▲▲▲▲▲▲|▲Birthdate▲▲▲▲▲▲▲|▲Phone▲▲▲▲▲▲▲▲▲▲▲]↵
[▲harper▲▲▲▲▲▲▲▲▲▲|▲2024-11-27▲▲▲▲▲▲|▲0887654321▲▲▲▲▲▲]↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲Hudson↵
Birthdate▲(YYYY▲MM▲DD):▲2023▲01▲10↵
Phone▲number:▲0815674321↵
===>▲Hint:▲Contact▲added↵
```

```
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲amelia↵
Birthdate▲(YYYY▲MM▲DD):▲2021▲08▲17↵
Phone▲number:▲0854321098↵
===>▲Hint:▲Contact▲added↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲Charlotte↵
Birthdate▲(YYYY▲MM▲DD):▲2022▲05▲22↵
Phone▲number:▲0921345678↵
===>▲Hint:▲Contact▲added↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲239naKOJ[↵
Birthdate▲(YYYY▲MM▲DD):▲al24▲ak▲3i↵
Phone▲number:▲cmao1-jc0j↵
===>▲Hint:▲Invalid▲name↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲Benjamin↵
Birthdate▲(YYYY▲MM▲DD):▲al24▲ak▲3i↵
Phone▲number:▲cmao1-jc0j↵
===>▲Hint:▲Invalid▲birthdate↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲Benjamin↵
Birthdate▲(YYYY▲MM▲DD):▲2023▲01▲10↵
Phone▲number:▲cmao1-jc0j↵
===>▲Hint:▲Invalid▲phone▲number↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲Benjamin↵
Birthdate▲(YYYY▲MM▲DD):▲2023▲01▲10↵
Phone▲number:▲0815674321↵
===>▲Hint:▲Contact▲added↵
Option▲>>▲l↵
Selected▲option:▲(l)↵
Found▲5▲records▲in▲total↵
[▲Name▲▲▲▲▲▲▲▲▲▲▲▲|▲Birthdate▲▲▲▲▲▲▲|▲Phone▲▲▲▲▲▲▲▲▲▲▲]↵
[▲Benjamin▲▲▲▲▲▲▲▲|▲2023-01-10▲▲▲▲▲▲|▲0815674321▲▲▲▲▲▲]↵
[▲Charlotte▲▲▲▲▲▲▲|▲2022-05-22▲▲▲▲▲▲|▲0921345678▲▲▲▲▲▲]↵
[▲Hudson▲▲▲▲▲▲▲▲▲▲|▲2023-01-10▲▲▲▲▲▲|▲0815674321▲▲▲▲▲▲]↵
[▲amelia▲▲▲▲▲▲▲▲▲▲|▲2021-08-17▲▲▲▲▲▲|▲0854321098▲▲▲▲▲▲]↵
[▲harper▲▲▲▲▲▲▲▲▲▲|▲2024-11-27▲▲▲▲▲▲|▲0887654321▲▲▲▲▲▲]↵
```

```
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Unable▲to▲add▲more▲contacts↵
Option▲>>▲u↵
Selected▲option:▲(u)↵
Name:▲Amelia↵
===>▲Hint:▲Contact▲not▲found↵
Option▲>>▲u↵
Selected▲option:▲(u)↵
Name:▲amelia↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Updating▲contact...↵
Name:▲Ga98&h1↵
Birthdate▲(YYYY▲MM▲DD):▲ka92▲H1▲c1↵
Phone▲number:▲an(*YNooa9↵
===>▲Hint:▲Invalid▲name↵
Option▲>>▲u↵
Selected▲option:▲(u)↵
Name:▲amelia↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Updating▲contact...↵
Name:▲Anna↵
Birthdate▲(YYYY▲MM▲DD):▲ka92▲H1▲c1↵
Phone▲number:▲an(*YNooa9↵
===>▲Hint:▲Invalid▲birthdate↵
Option▲>>▲u↵
Selected▲option:▲(u)↵
Name:▲amelia↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Updating▲contact...↵
Name:▲Anna↵
Birthdate▲(YYYY▲MM▲DD):▲1987▲12▲28↵
Phone▲number:▲an(*YNooa9↵
===>▲Hint:▲Invalid▲phone▲number↵
Option▲>>▲u↵
Selected▲option:▲(u)↵
Name:▲amelia↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Updating▲contact...↵
Name:▲Anna↵
Birthdate▲(YYYY▲MM▲DD):▲1987▲12▲28↵
Phone▲number:▲0344633949↵
===>▲Hint:▲Contact▲updated↵
Option▲>>▲l↵
Selected▲option:▲(l)↵
Found▲5▲records▲in▲total↵
[▲Name▲▲▲▲▲▲▲▲▲▲▲▲|▲Birthdate▲▲▲▲▲▲▲|▲Phone▲▲▲▲▲▲▲▲▲▲▲]↵
[▲Anna▲▲▲▲▲▲▲▲▲▲▲▲|▲1987-12-28▲▲▲▲▲▲|▲0344633949▲▲▲▲▲▲]↵
[▲Benjamin▲▲▲▲▲▲▲▲|▲2023-01-10▲▲▲▲▲▲|▲0815674321▲▲▲▲▲▲]↵
```

[▲Charlotte▲▲▲▲▲▲▲|▲2022-05-22▲▲▲▲▲▲|▲0921345678▲▲▲▲▲▲]↵
[▲Hudson▲▲▲▲▲▲▲▲▲▲|▲2023-01-10▲▲▲▲▲▲|▲0815674321▲▲▲▲▲▲]↵
[▲harper▲▲▲▲▲▲▲▲▲▲|▲2024-11-27▲▲▲▲▲▲|▲0887654321▲▲▲▲▲▲]↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵
Name:▲chalitte↵
===>▲Hint:▲Contact▲not▲found↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵
Name:▲Charlotte↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Contact▲deleted↵
Option▲>>▲a↵
Selected▲option:▲(a)↵
===>▲Hint:▲Adding▲contact...↵
Name:▲Marcus↵
Birthdate▲(YYYY▲MM▲DD):▲2004▲01▲30↵
Phone▲number:▲0911069051↵
===>▲Hint:▲Contact▲added↵
Option▲>>▲l↵
Selected▲option:▲(l)↵
Found▲5▲records▲in▲total↵
[▲Name▲▲▲▲▲▲▲▲▲▲▲|▲Birthdate▲▲▲▲▲▲▲|▲Phone▲▲▲▲▲▲▲▲▲▲]↵
[▲Anna▲▲▲▲▲▲▲▲▲▲▲|▲1987-12-28▲▲▲▲▲▲|▲0344633949▲▲▲▲▲▲]↵
[▲Benjamin▲▲▲▲▲▲▲|▲2023-01-10▲▲▲▲▲▲|▲0815674321▲▲▲▲▲▲]↵
[▲Hudson▲▲▲▲▲▲▲▲▲|▲2023-01-10▲▲▲▲▲▲|▲0815674321▲▲▲▲▲▲]↵
[▲Marcus▲▲▲▲▲▲▲▲▲|▲2004-01-30▲▲▲▲▲▲|▲0911069051▲▲▲▲▲▲]↵
[▲harper▲▲▲▲▲▲▲▲▲|▲2024-11-27▲▲▲▲▲▲|▲0887654321▲▲▲▲▲▲]↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵
Name:▲Anna↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Contact▲deleted↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵
Name:▲Marcus↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Contact▲deleted↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵
Name:▲Benjamin↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Contact▲deleted↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵

```
Name:▲Hudson↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Contact▲deleted↵
Option▲>>▲l↵
Selected▲option:▲(l)↵
Found▲1▲record▲in▲total↵
[▲Name▲▲▲▲▲▲▲▲▲▲▲▲|▲Birthdate▲▲▲▲▲▲▲|▲Phone▲▲▲▲▲▲▲▲▲▲▲]↵
[▲harper▲▲▲▲▲▲▲▲▲▲|▲2024-11-27▲▲▲▲▲▲|▲0887654321▲▲▲▲▲▲]↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵
Name:▲Harper↵
===>▲Hint:▲Contact▲not▲found↵
Option▲>>▲d↵
Selected▲option:▲(d)↵
===>▲Hint:▲Deleting▲contact...↵
Name:▲harper↵
===>▲Hint:▲Contact▲found↵
===>▲Hint:▲Contact▲deleted↵
Option▲>>▲l↵
Selected▲option:▲(l)↵
===>▲Hint:▲No▲records▲were▲found↵
Option▲>>▲q↵
Selected▲option:▲(q)↵
bye!↵
[3:23 user@ws hw]
```

1. 本題應繳交 Contact.c 與 Contact.h 兩個檔案，至於其他檔案則不需繳交。
2. 字串限制：
    1. 姓名不超過 15 個字元，此外僅接受大小寫字母及空格
    2. 年份不超過 4 個字元，僅接受數字
    3. 月份與日期皆不超過 2 個字元，僅接受數字
    4. 電話號碼不超過 10 個字元，僅接受數字
    5. 以上皆不會輸入空字串
3. 名字需依字典進行排序，順序為 **大寫** 到 **小寫**（A到Z至a到z

p4 字串替換

請參考如下 Main.c 的程式碼:

```
#include <stdio.h>
```

```
#include "ReplaceSubstring.h"

int main() {
    char article[2049], to_replace[16], replacement[16];

    scanf(" %2048[^\n]", article);
    scanf(" %16s", to_replace);
    scanf(" %16s", replacement);

    ReplaceSubstring(article, to_replace, replacement);

    printf("%s", article);

    return 0;
}
```

同學需完成並繳交 **ReplaceSubstring.c** 與 **ReplaceSubstring.h** 的 C 語言程式，其中分別包含 ReplaceSubstring() 函式的 實作（Implementation）與其 原型（Prototype）宣告。

本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c ReplaceSubstring.o
    gcc Main.c ReplaceSubstring.o

ReplaceSubstring.o: ReplaceSubstring.c ReplaceSubstring.h
    gcc -c ReplaceSubstring.c

clean:
    rm -f *.o *.out *.exe *.*~
```

本題可參考以下的執行結果：

```
[3:23 user@ws hw] ./a.out▲<▲testfile_1.txt↵
The▲CONDITIONS▲today▲is▲perfect.▲Sunny▲CONDITIONS▲lifts▲everyone's▲mood.▲This▲kind▲of▲CO
NDITIONS▲makes▲outdoor▲activities▲enjoyable.▲When▲the▲CONDITIONS▲is▲nice,▲parks▲are▲full.▲
Bad▲CONDITIONS▲keeps▲people▲indoors.▲We▲all▲prefer▲good▲CONDITIONS.▲If▲the▲CONDITIONS▲
stays▲like▲this,▲it'll▲be▲a▲great▲week.↵
[3:23 user@ws hw] ./a.out▲<▲testfile_2.txt↵
The▲City▲never▲sleeps.▲The▲City▲lights▲glow▲all▲night.▲In▲the▲City,▲people▲rush▲everywhere.▲Th
e▲heart▲of▲the▲City▲beats▲fast.▲Noise▲fills▲the▲City▲streets.▲Still,▲the▲City▲holds▲charm.▲Living▲i
n▲the▲City▲means▲constant▲motion▲and▲energy,▲but▲the▲City▲can▲feel▲lonely▲too.↵
[3:23 user@ws hw] ./a.out▲<▲testfile_3.txt↵
She▲loves▲to▲Bake.▲Every▲weekend,▲she▲will▲Bake▲something▲new.▲The▲smell▲of▲what▲she▲Bak
es▲fills▲the▲house.▲Her▲family▲gathers▲when▲she▲Bakes.▲To▲Bake▲is▲her▲passion.▲Some▲people
▲eat▲to▲live,▲but▲she▲lives▲to▲Bake.▲She▲dreams▲to▲Bake▲for▲the▲world▲someday.↵
[3:23 user@ws hw]
```

本題測試檔路徑:

- /home/stu/public/c2025s/c.hw4/p4/testfile_1.txt
- /home/stu/public/c2025s/c.hw4/p4/testfile_2.txt
- /home/stu/public/c2025s/c.hw4/p4/testfile_3.txt

1. 本次測試輸入 $0 \lt$ **段落長度** ≤ 1024 個字元。
2. 本題應繳交 ReplaceSubstring.c 與 ReplaceSubstring.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。

p5 * 五子棋-活三

程式演練 22 提供的程式碼提供了完整的程式碼判斷哪一方已經五子連線取得勝利。請參考課本 P13-18 的程式演練 22，修改程式碼增加功能，讓程式能夠找出棋盤中所有活三的位置。所謂「活三」是指已經完成三子連線的（縱、橫或斜向的連續三個相同顏色的棋子），且其前後兩端都沒有棋子。換句話說，這三顆棋子是若己方的可以選擇下子在活三的前後形成四子連線；若是對手的可以阻擋其形成四子連線。

為簡化規則，本題不需找出洞三，也就是連續的四個位置中，除頭尾外中間的某個位置留空，但是另外三子也為同一方的情況，當然洞三的頭尾外側也沒有對手防堵的棋子。

本題不限制 Turnin 所繳交的檔名，但必須繳交 Makefile 檔案並將可執行檔命名為 a.out且 Makefile 會使用到的檔案必須繳交，否則將導致編譯錯誤。

本題可參考以下的執行結果：

```
[3:23 user@ws hw] ./a.out▲<▲testfile_1.txt↵
Live▲three▲found:▲(F,▲▲2)▲(G,▲▲3)▲(H,▲▲4)↵
Live▲three▲found:▲(G,▲▲4)▲(H,▲▲4)▲(I,▲▲4)↵
Live▲three▲found:▲(L,▲▲6)▲(K,▲▲7)▲(J,▲▲8)↵
[3:23 user@ws hw] ./a.out▲<▲testfile_2.txt↵
Live▲three▲found:▲(M,▲▲4)▲(L,▲▲5)▲(K,▲▲6)↵
Live▲three▲found:▲(G,▲▲6)▲(H,▲▲6)▲(I,▲▲6)↵
Live▲three▲found:▲(M,▲14)▲(M,▲15)▲(M,▲16)↵
[3:23 user@ws hw] ./a.out▲<▲testfile_3.txt↵
Live▲three▲found:▲(L,▲▲6)▲(M,▲▲7)▲(N,▲▲8)↵
[3:23 user@ws hw] ./a.out▲<▲testfile_4.txt↵
Live▲three▲does▲not▲exist.↵
[3:23 user@ws hw]
```

請注意輸出棋盤位置的格式，也有可能會有兩位數的數字，請參考以下的結果：

```
[3:23 user@ws hw] ./a.out▲<▲testfile_XXXX.txt↵
Live▲three▲found:▲(F,▲12)▲(G,▲13)▲(H,▲14)↵
```

```
Live▲three▲found:▲(G,▲▲4)▲(H,▲▲4)▲(I,▲▲4)↵
[3:23 user@ws hw] ./a.out▲<▲testfile_2.txt↵
```

本題測試檔路徑：

- /home/stu/public/c2025s/c.hw4/p5/testfile_1.txt
- /home/stu/public/c2025s/c.hw4/p5/testfile_2.txt
- /home/stu/public/c2025s/c.hw4/p5/testfile_3.txt
- /home/stu/public/c2025s/c.hw4/p5/testfile_4.txt

1. 本題不限制 Turnin 所繳交的檔名，但必須繳交 Makefile 檔案並將可執行檔命名為 a.out。
2. Makefile 會使用到的檔案必須繳交，否則將導致編譯錯誤。
3. 批改時將採人工批改，無論判斷或輸出的順序是什麼，最終必須顯示以下兩個狀態：
    1. 輸出棋盤上每一組活三中三子於棋盤上的正確位置
    2. 若找不到活三則顯示 `Live▲three▲does▲not▲exist.↵`

【提示】建議同學完成學期末五子棋對決的相關程式碼時可以將不同的功能分類擺放，比如:

1. 與活三相關的功能可以將檔案命名為 「LiveThree.h」、「LiveThree.c」
2. 與死四相關的功能可以將檔案命名為 「DeadFour.h」、「DeadFour.c」
3. .....

最後，主程式透過 `#include “LiveThree.h”` 、`#include “DeadFour.h”`取得判斷下子所需要的函式。

p6 * 五子棋-死四

承上題並參考程式演練 22 提供的程式碼，修改程式碼增加功能，讓程式能夠找出棋盤中所有死四的位置。所謂「死四」是指已經完成四子連線的（縱、橫或斜向的連續四個相同顏色的棋子），且其前後有一端有對手的棋子防堵。換句話說，這四顆棋子是若己方的可以選擇下子在死四缺口形成五子連線並取得該回合勝利；若是對手的可以阻擋其形成五子連線。

同樣地，本題不需找出洞死四，也就是在連續的四個位置中，除頭尾外中間某個位置留空，但另外三子為同一方棋子的狀況，當然洞死四的頭尾已經有一端被對手防堵的棋子。

本題不限制 Turnin 所繳交的檔名，但必須繳交 Makefile 檔案並將可執行檔命名為 a.out，且 Makefile 會使用到的檔案必須繳交，否則將導致編譯錯誤。

本題可參考以下的執行結果：

```
[3:23 user@ws hw] ./a.out▲<▲testfile_1.txt↵
Dead▲four▲found:▲(H,▲11)▲(H,▲12)▲(H,▲13)▲(H,▲14)↵
Dead▲four▲found:▲(I,▲12)▲(J,▲12)▲(K,▲12)▲(L,▲12)↵
[3:23 user@ws hw] ./a.out▲<▲testfile_2.txt↵
Dead▲four▲does▲not▲exist.↵
[3:23 user@ws hw] ./a.out▲<▲testfile_3.txt↵
```

```
Dead▲four▲found:▲(L,▲▲7)▲(K,▲▲8)▲(J,▲▲9)▲(I,▲10)↵
[3:23 user@ws hw] ./a.out▲<▲testfile_4.txt↵
Dead▲four▲does▲not▲exist.↵
[3:23 user@ws hw]
```

本題測試檔路徑：

- 同上題測試檔

1. 本題不限制 Turnin 所繳交的檔名，但必須繳交 Makefile 檔案並將可執行檔命名為 a.out
2. Makefile 會使用到的檔案必須繳交，否則將導致編譯錯誤。
3. 批改時將採人工批改，無論判斷或輸出的順序是什麼，最終必須顯示以下兩個狀態：
    1. 輸出棋盤上每一組死四中四子於棋盤上的正確位置
    2. 若找不到死四則顯示 `Dead▲four▲does▲not▲exist.↵`