2024 NCKU Program Design I Homework 7

> 請不要嘗試攻擊 Judge 系統，否則此堂課將不予以通過
>
> Don't attack judge system otherwise you will fail this course.
>
> 一旦發現作業抄襲或是請人代寫之情形，學期作業成績將 "全部" 以 0 分計算
>
> One instances of severe plagiarism, hiring someone to write assignments, or similar activities are detected, the semester's assignment scores will be calculated as 0 point across the board.

> 對於作業有任何問題可以直接寄信到 f74114744@gs.ncku.edu.tw (mailto:f74114744@gs.ncku.edu.tw)，如果是其他課程上的問題需要處理請利用全體助教信箱 c2024@mail.csie.ncku.edu.tw (mailto:c2024@mail.csie.ncku.edu.tw)
>
> If you have any question about this homework tasks (ex. problem description), please feel free to contact me (資訊 115 陳俊安, f74114744@gs.ncku.edu.tw (mailto:f74114744@gs.ncku.edu.tw)).
>
> Otherwise, the common problem of this homework please send to TAs mail (c2024@mail.csie.ncku.edu.tw (mailto:c2024@mail.csie.ncku.edu.tw))

Before you start

- Make sure you can login the server by your personal account.

Submission

- Server IP: `140.116.246.48` , Port: `2024`

```
ssh 學號@140.116.246.48 -p 2024
```

- Login the system by your personal account. (Use the ssh command)
- Create an directory with name `HW7` in your home directory.
  - You can use the "pwd" command to confirm your current directory.

    - The "mkdir [name]" command can create a directory with the name [name]

- In `HW7` directory, you need to create 14 files with name "hw7-1.c", "hw7-1.h", "hw7-2.c", "hw7-2.h", "hw7-3.c", "hw7-3.h", "hw7-4.c", "hw7-4.h", "hw7-5.c", "hw7-5.h", "hw7-6.c", "hw7-7.c" and "hw7-7.h" respectively.
- You can directly use the command `hw7` to check whether the result of each question is correct.

Deadline: 2024/12/18 (23:59:59)

Homework 7 Information

A = B 是一款程式設計遊戲，此遊戲描述著作者因編譯系統 (Compiler) 課程忘記繳交作業而決定自己發明一個程式語言作為補交，遊戲中可以看到作者與教授之間的對話，也透過層層關卡讓使用者發現這個只有一個指令的程式語言可以創造出非常多無限可能，小至排序，大至將兩個二進位的數字相乘。

這款遊戲目前在 Steam 上得到了壓倒性的好評，是個資訊系學生適合遊玩的遊戲。某天就讀國立成功大學的小 C 在 Steam 上花了 102 元買了這個遊戲，並且現在希望你能夠利用程式設計 (一) 學到的知識將這一款遊戲簡單的實作出來！

A = B Rules

這一款遊戲只有一條指令可以使用，格式為 `string1 = string2` ，可以在同一份程式碼內打多行此格式的指令。此指令會針對輸入的字串 `input` 並將 `input` 中最左邊子字串為 `string1` 的字串替換成 `string2` 。以下舉幾個例子：

```
Input: abbc
Command: abb = bba
Output: bbac
```

```
Input: aaaaaab
Command: a = c
Output: caaaaab
```

備註：記得當有多個子字串為 `string1` 時，只會替換最左邊的那一個。

以下為 `A = B` 的執行的流程。

Step 1 從上到下找到第一個可以滿足要求的指令，並執行該指令。

Step 2 如果 **Step 1** 有找到可以執行的指令，重新回到 **Step 1**，否則，程式結束

以下我們舉一個例子：

Input

```
aaabbbbc
```

Command

```
a = b
b = c
ccc = NCKU
```

流程

```
aaabbbbc -> baabbbbc -> bbabbbbc -> bbbbbbbc -> cbbbbbbc ->
ccbbbbbc -> cccbbbbc -> ccccbbbc -> cccccbbc -> ccccccbc ->
cccccccc -> NCKUccccc -> NCKUNCKUcc
```

知道以上規則後，現在讓小 C 一起帶領你把這一個程式實作出來吧！首先我們先介紹一下這一個程式需要的功能。

- 程式開始執行後，歡迎使用者，並且讓使用者選擇關卡
- 請使用者輸入所有指令
- 生成測資，檢驗使用者的程式是否正確，並給予回饋
- 告訴使用者是否通過關卡

Subtask 1: 歡迎使用者，並讓使用者選擇關卡 - (10%, hw7-1.h, hw7-1.c)

首先，讓我們先從歡迎使用者開始吧！讓使用者自行決定要玩第一關還是第二關。

Function Requirements

Function: `int GameStart()`

請先輸出 `Welcome to the game A = B, please select the level` 。

並讓使用者輸入要玩哪一個關卡。這邊為了簡化問題，使用者只能輸入 `1` 或 `2` 。

如果輸入非這兩個內容必須回傳 `-1` ，否則直接回傳使用者選擇的關卡號碼。

> 保證輸入內容長度不超過 99，且最後一定會輸入 1 或 2，且保證內容不含有空格字元或空字串。

main.c

```c
#include <stdio.h>
#include "hw7-1.h"
int main() {
    int choose;
    while(1) {
        choose = GameStart();
        if( choose == -1 ) {
            printf("Please input 1 or 2 !\n");
        }
        else {
            printf("ok, you select the level %d\n", choose);
            break;
        }
    }
}
```

Sample (< Output, > Input)

```
< Welcome to the game A = B, please select the level
> 8
< Please input 1 or 2 !
< Welcome to the game A = B, please select the level
> NCKU
< Please input 1 or 2 !
< Welcome to the game A = B, please select the level
> 2
< ok, you select the level 2
```

Subtask 2 - 讓使用者輸入所有指令 (10%, p7-2.h, p7-2.c)

當使用者選擇完關卡後，我們必須讓使用者輸入所有滿足 `string1 = string2` 的指令。接下來請你提示使用者輸入指令，直到使用者輸入 `exit` 表示輸入完畢。

輸入完畢後，必須依序輸出使用者輸入的所有指令。

> 保證指令長度不超過 20，且指令總數不超過 50。
>
> 保證輸入的字串格式為 `string1 = string2`
>
> `string1` 與 `string2` 的長度保證不超過 8 且字串不含有空格字元。
>
> `string1` 保證不是空字串。

> `string2` 可以是空字串。不過即使 `string2` 是空字串我們依舊會讓格式保持為 `string1 = string2`

Function Requirements

Function: `char* getUserInput()`

回傳一個 char array 的 reference，提供給 `main.c` 中的 `strcpy` 使用，讓使用者該次輸入的指令可以被正確的複製到 `command[command_cnt]` 。

Hints: 如果你發現編譯器跳出來的警告與 stack memory 有關，請想想看某次助教課的時候教到當 Function 結束後，裡面的區域變數會發生什麼事？或許你就會知道問題在哪了！

Function: `void printAllInput()`

傳入 `command` 的 reference 與 `command_cnt` ，請透過這兩個參數輸出使用者輸入的所有指令。

main.c

```c
#include <stdio.h>
#include <string.h>
#include "hw7-2.h"

char command[51][21];
int main() {
    printf("Please input all the commands\n");
    int command_cnt = 0;
    while(1) {
        strcpy(command[command_cnt],getUserInput());
        if( strcmp(command[command_cnt++], "exit") == 0 ) {
            command_cnt--;
            printAllInput(command, command_cnt);
            break;
        }
    }

    return 0;
}
```

Sample (> Input, < Output)

```
< Please input all the commands
> a =
> a = b
> bb = aa
> NCKU = NCKU
> exit
< a =
< a = b
< bb = aa
< NCKU = NCKU
```

Subtask 3 - 整理指令 (10%, p7-3.h, p7-3.c)

> 特別注意：此題會需要 7-2 的 Function，如果 7-2 的 Function 是錯誤的，此題將無法通過

小 C 看到你的成果感到非常開心！不過他發現存在 `command` 的所有字串都是 `string1 = string2` 的形式，我們必須要將 `string1` 與 `string2` 提取出來。

接下來我們來把這個功能實作出來吧！

> 保證指令長度不超過 20，且指令總數不超過 50。
>
> 保證輸入的字串格式為 `string1 = string2`
>
> `string1` 與 `string2` 的長度保證不超過 8

Function Requirements

Function: `void ParsingData()`

依序傳入 `command[i]`, `string1[i]` 與 `string2[i]` 的 reference，並透過 **pass by reference** 的方式將結果儲存於 `string1[i]` 與 `string2[i]`。

main.c

```c
#include <stdio.h>
#include <string.h>
#include "hw7-3.h"
#include "hw7-2.h"

char string1[51][21], string2[51][21];
char command[51][21];
int main() {
    printf("Please input all the commands\n");
    int command_cnt = 0;
    while(1) {
        strcpy(command[command_cnt],getUserInput());
        if( strcmp(command[command_cnt++], "exit") == 0 ) {
            command_cnt--;
            break;
        }
    }

    for(int i=0;i<command_cnt;i++) {
        ParsingData(command[i], string1[i], string2[i]);
        printf("command %d: string1 = %s\n", i + 1,string1[i]);
        printf("command %d: string2 = %s\n", i + 1,string2[i]);
    }

    return 0;
}
```

Sample (> input, < output)

```
< Please input all the commands
> a = b
> NCKU = NCKUU
> bbb = aaa
> a =
> exit
< command 1: string1 = a
< command 1: string2 = b
< command 2: string1 = NCKU
< command 2: string2 = NCKUU
< command 3: string1 = bbb
< command 3: string2 = aaa
< command 4: string1 = a
< command 4: string2 =
```

Subtask 4 - 關卡設計 Level 1 (10%, p7-4.h, p7-4.c)

使用者指令的部分暫時告一個段落了！接下來我們要來設計關卡。

小 C 決定將第一關設計為將字串中連續的 `a` 刪除，我們先設計一個 Function 能夠針對測試資料得到正確答案。

> 保證 `testcase` 的長度不超過 50 且不含有空格字元也不是空字串。

Function Requirements

Function: `char* getAnswer1()`

傳入字串 `testcase` 的 reference，並回傳將 `testcase` 連續的 `a` 刪除後的結果提供給 `strcpy` 複製。

main.c

```c
#include <stdio.h>
#include <string.h>
#include "hw7-4.h"
int main() {
    char testcase[51] = {};
    scanf("%s", testcase);
    char answer[51] = {};
    strcpy(answer, getAnswer1(testcase));
    printf("%s\n", answer);

    return 0;
}
```

Sample Input 1

```
aaaaab
```

Sample Output 1

```
b
```

Sample Input 2

```
abbaaabbaaba
```

Sample Output 2

```
abbbbba
```

Subtask 5 - 關卡設計 Level 2 (10%, p7-5.h, p7-5.c)

接下來我們一起來設計第二個關卡，第二個關卡為將字串排序，讓我們也寫一個 Function 來得到測試資料的正確答案。

> 保證 `testcase` 的長度不超過 50 且不含有空格字元也不是空字串。

Function Requirements

Function: `char* getAnswer2()`

傳入 `testcase` 的 reference 並回傳一個 reference 提供給 `strcpy` 複製將 `testcase` 依照字典序 (ASCII 大小) 由小到大排好的結果。

Hints: 你可以上網尋找任何排序方法，或是自己先學看看 qsort 怎麼使用

main.c

```c
#include <stdio.h>
#include <string.h>
#include "hw7-5.h"

int main() {
    char testcase[51] = {};
    scanf("%s", testcase);
    char answer[51] = {};
    strcpy(answer, getAnswer2(testcase));
    printf("%s\n", answer);

    return 0;
}
```

Sample Input 1

```
ccbbaa
```

Sample Output 1

```
aabbcc
```

Subtask 6 - 執行使用者的指令 (10%, hw7-6.h, hw7-6.c)

接下來，我們回到與使用者有關的部分。

讓我們來設計一個 Function 試著執行使用者的指令。

> 指令限制與 Subtask 2 一樣，且會使用到 Subtask 2 與 Subtask 3 的 Function (會 include hw7-2.h, hw7-3.h)
>
> 另外，保證最終結果的字串長度與操作的中途字串長度不會超過 50，且指令不會有無窮迴圈的情況。

Function Requirements

Function: `char* ProcessingCommand()`

傳入 `string1` 與 `string2` 的 reference，與 `testcase` 還有 `command_cnt` 與 `test_mode`，如果 `test_mode = 1` 則請在該 Function 依序輸出每一次執行完某個指令後的結果。

並且將最後的結果回傳一個 reference 提供給 `strcpy` 複製。

> 請務必查看作業說明一開始的執行流程說明！！！

main.c

```c
#include <stdio.h>
#include <string.h>
#include "hw7-6.h"
#include "hw7-3.h"
#include "hw7-2.h"

char string1[51][21], string2[51][21];
char command[51][21];
int main() {
    printf("Please input all the commands\n");
    int command_cnt = 0;
    while(1) {
        strcpy(command[command_cnt],getUserInput());
        if( strcmp(command[command_cnt++], "exit") == 0 ) {
            command_cnt--;
            break;
        }
    }

    for(int i=0;i<command_cnt;i++) {
        ParsingData(command[i], string1[i], string2[i]);
    }
    printf("Pleas input the testcase:\n");
    char testcase[51] = {};
    scanf("%s", testcase);
    char output[51] = {};
    strcpy(output, ProcessingCommand(string1, string2, testcase, command_cnt, 1));
    printf("Output: %s\n", output);

    return 0;
}
```

Sample (> input, < output)

```
< Please input all the commands
> a = b
> b = c
> ccc = NCKU
> exit
< Pleas input the testcase:
> aaabbbbc
< baabbbbc
< bbabbbbc
< bbbbbbbc
< cbbbbbbc
< ccbbbbbc
< cccbbbbc
< ccccbbbc
< cccccbbc
< ccccccbc
< cccccccc
< NCKUccccc
< NCKUNCKUcc
< Output: NCKUNCKUcc
```

Subtask 7 - 生成測資 **(10%, hw7-7.h, hw7-7.c)**

接下來我們要生成一些測資來驗證使用者是否通過關卡。

為了簡化問題，我們將測試資料限定只會出現 `a`, `b` 和 `c` 這三種字元其中一種。

接下來請你設計一個 Function，傳入指定長度，與目標陣列 `all_testcase` 。

透過 **pass by reference** 的方式將所有排列組合按照字典順序 (ASCII 大小) 依序儲存在 `all_testcase` 。

Hints 1: 這題是想要考驗你會不會使用遞迴，建議不熟悉的話可以先從畫出遞迴樹下手。

Hints 2: 雖然你不能更改 `main.c` ，但 `hw7-7.c` 除了指定的 Function 是必要的之外，你也是可以自己新增任何你想要寫的 Function 作為輔助。

Function Requirements

Function: `void GeneratingTest()`

傳入指定長度 `target_len` 與目標陣列 `all_testcase` 的 reference，將所有排列組合透過 pass by reference 的方式將結果儲存於 `all_testcase` 陣列中。

> 保證 `target_len` 為正整數且介於 $[1, 10]$

main.c

```c
#include <stdio.h>
#include "hw7-7.h"

char all_testcase[59049][11]; // 59049 = 3^10

int main() {
    int target_len;
    scanf("%d", &target_len);
    GeneratingTest(target_len, all_testcase);

    int total = 1;
    for(int i=0;i<target_len;i++) total *= 3;
    for(int i=0;i<total;i++) {
        for(int j=0;j<target_len;j++) printf("%c", all_testcase[i][j]);
        printf("\n");
    }

    return 0;
}
```

Sample Input 1

```
2
```

Sample Output 1

```
aa
ab
ac
ba
bb
bc
ca
cb
cc
```

Sample Input 2

```
3
```

Sample Output 2

```
aaa
aab
aac
aba
abb
abc
aca
acb
acc
baa
bab
bac
bba
bbb
bbc
bca
bcb
bcc
caa
cab
cac
cba
cbb
cbc
cca
ccb
ccc
```

Subtask 8 - 遊戲大功告成！ **(30%,** 不需要創建 **hw7-8** 相關檔案**)**

恭喜你！接下來我們只要將之前你寫的所有 Function 整合一下就有一個簡單的小遊戲囉！

小 C 已經幫你們寫好了，就直接跑跑看下面這一個程式吧！(你也可以自己玩看看)

如果前面寫的 Function 都是沒問題的就會直接拿到分數了唷 ><

小 C 就幫你到這邊了，我們下一次作業再見囉！

> 請注意，本題為綜合測試題，目的為綜合測試所有 Function 功能正常

main.c

```c
#include <stdio.h>
#include <string.h>
#include "hw7-1.h"
#include "hw7-2.h"
#include "hw7-3.h"
#include "hw7-4.h"
#include "hw7-5.h"
#include "hw7-6.h"
#include "hw7-7.h"
char all_testcase[59049][11];
char string1[51][21], string2[51][21];
char command[51][21];
int main() {
    int choose;
    while(1) {
        choose = GameStart();
        if( choose == -1 ) {
            printf("Please input 1 or 2 !\n");
        }
        else {
            printf("ok, you select the level %d\n", choose);
            break;
        }
    }

    printf("Please input all the commands\n");
    int command_cnt = 0;
    while(1) {
        strcpy(command[command_cnt],getUserInput());
        if( strcmp(command[command_cnt++], "exit") == 0 ) {
            command_cnt--;
            break;
        }
    }

    for(int i=0;i<command_cnt;i++) {
        ParsingData(command[i], string1[i], string2[i]);
    }

    int target_len, total = 1;
    printf("Please input the length of testcase\n");
    scanf("%d", &target_len);
    GeneratingTest(target_len, all_testcase);
    for(int i=0;i<target_len;i++) total *= 3;
    for(int i=0;i<total;i++) {
        printf("==============================Testcase %d=======================
        printf("Input Data: %s\n", all_testcase[i]);
        char output[51] = {};
        strcpy(output, ProcessingCommand(string1, string2, all_testcase[i], comman
        char answer[51] = {};
        if( choose == 1 ) {
            strcpy(answer, getAnswer1(all_testcase[i])); // level 1
        }
```

```
53          }
54          else {
55              strcpy(answer, getAnswer2(all_testcase[i])); // level 2
56          }
57
58          printf("Player Result: %s\n", output);
59          printf("Answer: %s\n", answer);
60          if( strcmp(output, answer) == 0 ) printf("Result: Accepted\n");
61          else printf("Result: Wrong Answer\n");
62      }
63
64      return 0;
65  }
```

Sample 1 (這邊只特別把 **input** 用 **>** 標記出來**)**

```
Welcome to the game A = B, please select the level
> 1
ok, you select the level 1
Please input all the commands
> a =
> exit
Please input the length of testcase
> 3
==============================Testcase 1==============================
Input Data: aaa
Player Result:
Answer:
Result: Accepted
==============================Testcase 2==============================
Input Data: aab
Player Result: b
Answer: b
Result: Accepted
==============================Testcase 3==============================
Input Data: aac
Player Result: c
Answer: c
Result: Accepted
==============================Testcase 4==============================
Input Data: aba
Player Result: b
Answer: aba
Result: Wrong Answer
==============================Testcase 5==============================
Input Data: abb
Player Result: bb
Answer: abb
Result: Wrong Answer
==============================Testcase 6==============================
Input Data: abc
Player Result: bc
Answer: abc
Result: Wrong Answer
==============================Testcase 7==============================
Input Data: aca
Player Result: c
Answer: aca
Result: Wrong Answer
==============================Testcase 8==============================
Input Data: acb
Player Result: cb
Answer: acb
Result: Wrong Answer
==============================Testcase 9==============================
Input Data: acc
Player Result: cc
Answer: acc
```

```
Result: Wrong Answer
==============================Testcase 10==============================
Input Data: baa
Player Result: b
Answer: b
Result: Accepted
==============================Testcase 11==============================
Input Data: bab
Player Result: bb
Answer: bab
Result: Wrong Answer
==============================Testcase 12==============================
Input Data: bac
Player Result: bc
Answer: bac
Result: Wrong Answer
==============================Testcase 13==============================
Input Data: bba
Player Result: bb
Answer: bba
Result: Wrong Answer
==============================Testcase 14==============================
Input Data: bbb
Player Result: bbb
Answer: bbb
Result: Accepted
==============================Testcase 15==============================
Input Data: bbc
Player Result: bbc
Answer: bbc
Result: Accepted
==============================Testcase 16==============================
Input Data: bca
Player Result: bc
Answer: bca
Result: Wrong Answer
==============================Testcase 17==============================
Input Data: bcb
Player Result: bcb
Answer: bcb
Result: Accepted
==============================Testcase 18==============================
Input Data: bcc
Player Result: bcc
Answer: bcc
Result: Accepted
==============================Testcase 19==============================
Input Data: caa
Player Result: c
Answer: c
Result: Accepted
==============================Testcase 20==============================
Input Data: cab
```

```
Player Result: cb
Answer: cab
Result: Wrong Answer
==============================Testcase 21==============================
Input Data: cac
Player Result: cc
Answer: cac
Result: Wrong Answer
==============================Testcase 22==============================
Input Data: cba
Player Result: cb
Answer: cba
Result: Wrong Answer
==============================Testcase 23==============================
Input Data: cbb
Player Result: cbb
Answer: cbb
Result: Accepted
==============================Testcase 24==============================
Input Data: cbc
Player Result: cbc
Answer: cbc
Result: Accepted
==============================Testcase 25==============================
Input Data: cca
Player Result: cc
Answer: cca
Result: Wrong Answer
==============================Testcase 26==============================
Input Data: ccb
Player Result: ccb
Answer: ccb
Result: Accepted
==============================Testcase 27==============================
Input Data: ccc
Player Result: ccc
Answer: ccc
Result: Accepted
```

Sample 2

```
Welcome to the game A = B, please select the level
> 2
ok, you select the level 2
Please input all the commands
> ba = ab
> cb = bc
> ca = ac
> exit
Please input the length of testcase
> 3
================================Testcase 1================================
Input Data: aaa
Player Result: aaa
Answer: aaa
Result: Accepted
================================Testcase 2================================
Input Data: aab
Player Result: aab
Answer: aab
Result: Accepted
================================Testcase 3================================
Input Data: aac
Player Result: aac
Answer: aac
Result: Accepted
================================Testcase 4================================
Input Data: aba
Player Result: aab
Answer: aab
Result: Accepted
================================Testcase 5================================
Input Data: abb
Player Result: abb
Answer: abb
Result: Accepted
================================Testcase 6================================
Input Data: abc
Player Result: abc
Answer: abc
Result: Accepted
================================Testcase 7================================
Input Data: aca
Player Result: aac
Answer: aac
Result: Accepted
================================Testcase 8================================
Input Data: acb
Player Result: abc
Answer: abc
Result: Accepted
================================Testcase 9================================
Input Data: acc
```

```
Player Result: acc
Answer: acc
Result: Accepted
==============================Testcase 10==============================
Input Data: baa
Player Result: aab
Answer: aab
Result: Accepted
==============================Testcase 11==============================
Input Data: bab
Player Result: abb
Answer: abb
Result: Accepted
==============================Testcase 12==============================
Input Data: bac
Player Result: abc
Answer: abc
Result: Accepted
==============================Testcase 13==============================
Input Data: bba
Player Result: abb
Answer: abb
Result: Accepted
==============================Testcase 14==============================
Input Data: bbb
Player Result: bbb
Answer: bbb
Result: Accepted
==============================Testcase 15==============================
Input Data: bbc
Player Result: bbc
Answer: bbc
Result: Accepted
==============================Testcase 16==============================
Input Data: bca
Player Result: abc
Answer: abc
Result: Accepted
==============================Testcase 17==============================
Input Data: bcb
Player Result: bbc
Answer: bbc
Result: Accepted
==============================Testcase 18==============================
Input Data: bcc
Player Result: bcc
Answer: bcc
Result: Accepted
==============================Testcase 19==============================
Input Data: caa
Player Result: aac
Answer: aac
Result: Accepted
```

```
==============================Testcase 20==============================
Input Data: cab
Player Result: abc
Answer: abc
Result: Accepted
==============================Testcase 21==============================
Input Data: cac
Player Result: acc
Answer: acc
Result: Accepted
==============================Testcase 22==============================
Input Data: cba
Player Result: abc
Answer: abc
Result: Accepted
==============================Testcase 23==============================
Input Data: cbb
Player Result: bbc
Answer: bbc
Result: Accepted
==============================Testcase 24==============================
Input Data: cbc
Player Result: bcc
Answer: bcc
Result: Accepted
==============================Testcase 25==============================
Input Data: cca
Player Result: acc
Answer: acc
Result: Accepted
==============================Testcase 26==============================
Input Data: ccb
Player Result: bcc
Answer: bcc
Result: Accepted
==============================Testcase 27==============================
Input Data: ccc
Player Result: ccc
Answer: ccc
Result: Accepted
```