國立屏東大學 資訊工程系 程式設計(二)

Turnin作業2

- Turnin Code: c.hw2
- Due Date: 3/12 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hw2，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hw2資料夾，使用以下指令將其上傳：

[user@ws hw2]$ turnin␣c.hw2␣p1↵

當然，你也可以等到所有題目都完成後，再回到hw1資料夾，使用以下指令將所有題目都加以上傳：

[user@ws hw2]$ turnin␣c.hw2␣.↵



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 hw2]$ turnin -ls c.hw2
.:
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

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

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

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

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

./p5:
total 8
-rw-rw----. 1 turninman 1669401585 5 Mar  5 20:57 IsPalindrome.c
-rw-rw----. 1 turninman 1669401585 5 Mar  5 20:57 IsPalindrome.h

[user@ws hw2]$
```

【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 hw2]$ `turnin▲-rm▲c.hw2↵`

---

p1 字串長度

請參考下面的Main.c程式:

```
#include <stdio.h>

#include "GetStringLength.h"

int main() {
    char src[2048];

    printf("Enter a string: ");
    scanf("%[^\n]", src);

    printf("length: %d\n", GetStringLength(src));

    return 0;
```

```
}
```

你必須完成名為 GetStringLength.c 與 GetStringLength.h 的C語言程式，其中分別包含 GetStringLength() 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 GetStringLength.o
    gcc Main.c GetStringLength.c
GetStringLength.o: GetStringLength.c
    gcc -c GetStringLength.c

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

此題執行時可讓使用者輸入一個不超過2047個字元的字串(遇到Enter鍵即結束輸入)，計算字串中**包含多少個英文大小寫字母**後加以輸出。可參考以下的執行結果：

```
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲Hello,▲World!↵
length:▲10↵
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲methiony▲lthreony▲lthreonylglutaminylarginyliso
leucine▲▲▲▲▲▲▲▲▲▲▲▲▲▲↵
length:▲52↵
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲Apple▲iPhone▲16e▲128GB.↵
length:▲14↵
[user@ws hw]$
```



1. 本題 **所有函式** 內容需在 **GetStringLength.c** 檔案進行實作，否則將不予計分。
2. 本題應繳交 GetStringLength.c 與 GetStringLength.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。

p2 日期格式

請參考下面的Main.c程式:

```
#include <stdio.h>

#include "DateFormat.h"

int main() {
    int year, month, day;
    char type[3];
    printf("Date: ");
    scanf(" %d/%d/%d", &year, &month, &day);
    if (!CheckDate(year, month, day)) {
        printf("Error: Invalid date value\n");
        return 0;
    }
    printf("Format Type: ");
    scanf(" %s", type);
    if (!CheckRegionType(type)) {
        printf("Error: Invalid region format\n");
        return 0;
    }
    DateFormat(year, month, day, type);
}
```

你必須完成名為 DateFormat.c 與 DateFormat.h 的C語言程式，其中分別包含 DateFormat()、CheckDate() 以及 CheckRegionType() 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 DateFormat.o
    gcc Main.c DateFormat.o
DateFormat.o: DateFormat.c
    gcc -c DateFormat.c

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

此題的執行結果如下：

```
[user@ws hw]$ ./a.out↵
Date:▲2017/3/11↵
Format▲Type:▲US↵
The▲date▲“March 11th,▲2017”▲is▲in▲[US]▲format↵
[3:23 user@ws hw] ./a.out↵
Date:▲2028/12/21↵
Format▲Type:▲UK↵
The▲date▲“21st▲December,▲2028”▲is▲in▲[UK]▲format↵
[user@ws hw]$ ./a.out↵
Date:▲2004/01/30↵
```

```
Format▲Type:▲TW↵
The▲date▲"2004年1月30日"▲is▲in▲[TW]▲format↵
[user@ws hw]$ ./a.out↵
Date:▲2025/5/40↵
Error:▲Invalid▲date▲value↵
[user@ws hw]$ ./a.out↵
Date:▲2024/11/12↵
Format▲Type:▲ZZ↵
Error:▲Invalid▲region▲format↵
[user@ws hw]$
```



1. 本題 **所有函式** 內容需在 **DateFormat.c** 檔案進行實作，否則將不予計分。
2. 本題應繳交 DateFormat.c 與 DateFormat.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。
3. 本題出現符號如□**"**□等，皆為 「**半形**」□
4. 本題關於在輸入`Format▲Type:▲`時，只有**"US"**□**"UK"和 " TW"三個選項(皆為大寫)**，輸入其他數值視為錯誤，輸出`Error:▲Invalid▲region▲format`□
5. 本題年月日數值範圍如下：
   - $year > 0$
   - $month \ge 1 , month \le 12$
   - $day \ge 1 , day \le 31$
   - 超出範圍將輸出`Error:▲Invalid▲date▲value`

p3 大海撈針

請參考下面的Main.c程式:

```
#include <stdio.h>

#include "GetCountsOfSubString.h"

int main() {
    char srcstr[4097], substr[129];
    int count;

    printf("Enter the source string: ");
    scanf("%4096[^\n]", srcstr);
    printf("Enter the substring: ");
    scanf("%128s", substr);

    count = GetCountsOfSubString(srcstr, substr);
```

```
    switch (count) {
        case 0:
            printf("The needle \"%s\" does not appear in the haystack.\n",
substr);
            break;
        case 1:
            printf("The needle \"%s\" appears %d time in the haystack.\n",
substr, count);
            break;
        default:
            printf("The needle \"%s\" appears %d times in the haystack.\n",
substr, count);
    }

    return 0;
}
```

你必須完成名為 GetCountsOfSubString.c 與 GetCountsOfSubString.h 的C語言程式，其中分別包含 GetCountsOfSubString() 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 GetCountsOfSubString.o
    gcc Main.c GetCountsOfSubString.o
GetCountsOfSubString.o: GetCountsOfSubString.c
    gcc -c GetCountsOfSubString.c

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

本題有提供測試檔方便同學們測試，其路徑如下：

```
/home/stu/public/c2025s/c.hw2/p3/testfile_1.txt

/home/stu/public/c2025s/c.hw2/p3/testfile_2.txt

/home/stu/public/c2025s/c.hw2/p3/testfile_3.txt
```

同學可以透過指令將上述測試檔複製到自己程式的資料夾下進行測試。這些測試檔案的第1行包含有一個要搜尋的字串，我們將其稱為needle第2行則是一個haystack(長度最多可達4096個字元的字串)。此題將要從haystack裡，找出needle出現多少次！要注意的是，在haystack裡尋找needle將不區分大小寫英文字母。

此題的執行結果如下：

```
[3:23 user@ws hw] ./a.out < testfile_1.txt↵
The▲needle▲"Wake"▲appears▲5▲times▲in▲the▲haystack.↵
[3:23 user@ws hw] ./a.out < testfile_2.txt↵
```

```
The▲needle▲"Matter"▲appears▲1▲time▲in▲the▲haystack.↵
[3:23 user@ws hw] ./a.out < testfile_3.txt↵
The▲needle▲"blanket"▲does▲not▲appear▲in▲the▲haystack.↵
[3:23 user@ws hw]
```



1. 本題 **所有函式** 內容需在 **GetCountsOfSubString.c** 檔案進行實作，否則將不予計分。
2. 本題應繳交 GetCountsOfSubString.c 與 GetCountsOfSubString.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。

p4 身分證字號驗證器(*)

假設中華民國身分證字號中有其中幾碼是有特別的含義，從左到右依序定義第一碼直到最右邊的第十碼，其中：

- 第一碼代表申報戶口所在的縣市（戶籍地），如下表所示。
- 第二碼的值：1代表男性，2代表女性
- 第三碼：
  1. 0-5：台灣出生的本國國民
  2. 6：原為外國人之入籍國民
  3. 7：原無國籍國民
  4. 8：原為香港或澳門居民的入籍國民
  5. 9：原為大陸居民的入籍國民

代號	轉換數值	縣市	代號	轉換數值	縣市	代號	轉換數值	縣市	代號	轉換數值	縣市	代號	轉換數值	縣市
A	10	台北市	G	16	宜蘭縣	N	22	彰化縣	V	29	台東縣	L	20	台中縣
B	11	台中市	H	17	桃園市	O	35	新竹市	W	32	金門縣	R	25	台南縣
C	12	基隆市	I	34	嘉義市	P	23	雲林縣	X	30	澎湖縣	S	26	高雄縣
D	13	台南市	J	18	新竹縣	Q	24	嘉義縣	Z	33	連江縣	Y	31	陽明山管理局
E	14	高雄市	K	19	苗栗縣	T	27	屏東縣						
F	15	新北市	M	21	南投縣	U	28	花蓮縣	註：灰色字體為不再配賦之代碼					

計算身分證字號是否有效，我們先將使用者輸入的身分證字號定義為 $LD_1D_2D_3D_4D_5D_6D_7D_8D_9$□例如 "A123456789" 的$L = $'A'; $D_1 = 1$; $D_2 = 2$ ... 以此類推。接著我們必須將 L 置換成其戶籍地的轉換數值，並將轉換數值中十位數與個位數的數值分別放入 L_{D_1} 與 L_{D_2}□$L_{D_1}=1,L_{D_2}=0$□□ 最後，帶入以下驗證公式：

$$ Sum = L_{D_1}\times 1 + L_{D_2}\times 9 + D_1\times 8 + D_2\times 7 + D_3\times 6 +D_4\times 5 +D_5\times 4 +D_6\times 3 +D_7\times 2 +D_8\times 1 + D_9 \times 1 $$

若 Sum 能夠被 10 整除，該身分證字號為有效，反之為無效，請參考下面的 Main.c 程式:

```
#include <stdio.h>
#include <string.h>
#include "CheckID.h"
#define length 11
#define true 1
#define false 0
int main(){
    char str[length];
    char output[200];
    printf("請輸入身分證號碼: ");
    scanf("%[A-Z0-9]", str);
    if(formatCheck(str) == false){
        printf("格式錯誤!\n");
        return 0;
    }
    if(isValid(str) == false){
        printf("身分證號碼無效!\n");
        return 0;
    }
    strcat(output, "所輸入的身分證號碼: [");
    strcat(output, str);
    strcat(output, "]\n");
    strcat(output, "戶籍地: ");
    char * p =  &str[0];
    strcatHouseholdReg(p, output);
    p = &str[1];
    strcat(output, "性別: ");
    strcat(output, whichGender(p));
    strcat(output, "\n");
    p = &str[2];
    strcat(output, "身分別: ");
    strcat(output, whichIdentityType(p));
    printf("%s", output);
}
```

你必須完成名為 CheckID.c與 CheckID.h 的C語言程式，並實作 formatCheck()、isValid()、strcatHouseholdReg()、whichGender() 與 whichIdentityType() 函式，本題的相關程式將使用以下的Makefile進行編譯：

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

此題的執行結果如下：

```
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼:  A123456789↵
所輸入的身分證號碼:▲[A123456789]↵
戶籍地:  台北市
性別:  男性
身分別:  台灣出生的本國國民
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼:  Y161260320↵
所輸入的身分證號碼:▲[Y161260320]↵
戶籍地:  陽明山管理局
性別:  男性
身分別:  原為外國人之入籍國民
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼:  F185593126↵
所輸入的身分證號碼:▲[F185593126]↵
戶籍地:  新北市
性別:  男性
身分別:  原為香港或澳門居民的入籍國民
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼:  E285593181↵
所輸入的身分證號碼:▲[E285593181]↵
戶籍地:  高雄市
性別:  女性
身分別:  原為香港或澳門居民的入籍國民
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼:  C341725924↵
格式錯誤!
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼:  f185593126↵
格式錯誤!
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼:  E152847231↵
身分證號碼無效!
[3:23 user@ws hw] ./a.out↵
請輸入身分證號碼: E2595734↵
格式錯誤!
[3:23 user@ws hw]
```



1. 本題 **所有函式** 內容需在 **CheckID.c** 檔案進行實作，否則將不予計分。
2. 本題應繳交 CheckID.c與 CheckID.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。

p5 判斷迴文(*)

迴文是指依由左至右或由右至左的順序閱讀，其內容皆相同的字串。例如單字Level與「Madam以及句子Borrow or rob?與「Was it a car or a cat I saw?皆屬之。請設計C語言程式 IsPalindrome.c 以及其

標頭檔 IsPalindrome.h，設計一個名為 IsPalindrome() 的函式，並搭配下列的Main.c程式讓使用者輸入一個長度不超過256個字元的字串(遇到Enter鍵即結束輸入)，並判斷是否為迴文(Palindrome)後輸出結果：

```
#include <stdio.h>

#include "IsPalindrome.h"

int main() {
    char str[257];
    for (int i = 0; i < 257; i++) {
    str[i] = 0;
    }
    printf("Enter a string: ");
    scanf("%256[^\n]", str);

    if (IsPalindrome(str)) {
        printf("\"%s\" is a palindrome.\n", str);
    } else {
        printf("\"%s\" is not a palindrome.\n", str);
    }

    return 0;
}
```

你必須完成名為 IsPalindrome.c 與 IsPalindrome.h 的C語言程式，其中分別包含 IsPalindrome() 函式的 **實作(Implementation)**與其 **原型(Prototype)**宣告。

```
all: Main.c IsPalindrome.o
    gcc Main.c IsPalindrome.o
IsPalindrome.o: IsPalindrome.c
    gcc -c IsPalindrome.c

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

請特別注意，我們在進行迴文判讀時，將不區分大小寫英文字母，並將阿拉伯數字、標點符號、空白字元與特殊符號等皆加以去除。例如(Borrow 100 or rob 1000?)在判讀時，可以先將所有英文字母轉為大寫，並去除空白、問號、數字後，將原字串視為(BORROWORROB)再進行判讀；由於判讀內容依由左至右以及由右至左的順序閱讀皆相同，因此(Borrow 100 or rob 1000?)將被判讀為迴文。

此題的執行結果如下：

```
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲Madam↵
“Madam”▲is▲a▲palindrome.↵
```

```
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲rEgigEx↵
“rEgigEx” is not a palindrome.↵
[user@ws hw]$ ./a.out↵
Enter▲a string:▲↵
””▲is▲not▲a▲palindrome.↵
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲A▲Santa▲at▲NASA!↵
“A▲Santa▲at▲NASA!”▲is▲a▲palindrome.↵
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲666▲999▲666↵
“666▲999▲666”▲is▲not▲a▲palindrome.↵
[user@ws hw]$ ./a.out↵
Enter▲a▲string:▲Borrow▲100▲or▲rob▲1000?↵
“Borrow▲100▲or▲rob▲1000?”▲is▲a▲palindrome.↵
[user@ws hw]$
```



1. 本題 **所有函式** 內容需在 **IsPalindrome.c** 檔案進行實作，否則將不予計分。
2. 本題應繳交 IsPalindrome.c 與 IsPalindrome.h 兩個檔案，至於 Main.c 與 Makefile 則不需繳交。

From:
https://junwu.nptu.edu.tw/dokuwiki/ - **Jun Wu的教學網頁**
國立屏東大學資訊工程學系
CSIE, NPTU
Total: 333963

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2025spring:hw2

Last update: **2025/03/12 15:06**