國立屏東大學 資訊工程系 程式設計(一)

Turnin作業6

- 範圍：chapter 14
- Turnin Code: c.hw6
- Due Date: 2026/05/20 23:59 **Hard Deadline**

繳交方式說明

本次作業繳交將以資料夾的形式繳交，需要為每一題建立一個資料夾（資料夾名稱為該題題目前方之代號，第一題為：p1，第二題為：p2，餘以此類推），

繳交說明可參考作業3：**連結**

任何未依照正確繳交格式的檔案將以 0 分計算



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！

---

p1 數字箱

在處理數量不確定的整數資料時，傳統固定長度的陣列容易發生空間不足或浪費空間的問題。因此衍生出動態陣列的方式專門處理對於資料不固定的情況。

本題要求實作一個C語言動態數字箱，使用者會先輸入箱子的初始大小，若 **初始大小 $\lt$ 1** 時，程式會自動以 **INIT_CAP(定義在 define.h)** 的大小設置初始空間，緊接著使用者將會輸入一連串的整數，直到某輸入的內容不是數字為止，當箱子空間不足時，它能向系統以 **當前 capacity * GROWTH_RATE(定義在 define.h)** 的方式申請更大的空間並保留原有的數據。



實作注意

- 本題假設 **所有的輸入資料皆正確**，因此請同學不用額外處理輸入等問題。

請參考以下 define.h 與 main.c 的內容：

```
#define INIT_SIZE 2
#define GROWTH_RATE 2

typedef struct
{
    int* data;
    unsigned int size;
    unsigned int capacity;
} numBox;
```

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

#include "numBox.h"

int main()
{
    int cap = 0;
    printf("init capacity: ");
    scanf(" %d", &cap);

    numBox* nb = numBoxNew(cap);

    int data = 0;
    printf("input: ");
    while (scanf(" %d", &data) == 1)
    {
        numBoxAdd(nb, data);
    }

    printf("\n");
    numBoxList(nb);
    numBoxFree(nb);
    return 0;
}
```

本題需請同學定義並實作下列函式：

- **numBoxNew()**
- **numBoxFree()**
- **numBoxAdd()**
- **numBoxList()**

完成後，將內容放入以下檔案：

- **numBox.c**：此文件需包含上述 **函式** 的實作。
- **numBox.h**：此文件需包含上述 **函式** 的定義。

另外，本題目使用以下 Makefile 進行批改：

```
all: main.c numBox.o
    gcc main.c numBox.o
numBox.o: numBox.c
    gcc -c numBox.c
clean:
    rm -f *.*~ a.* *.o *.o~
```

本題的執行結果可參考如下：

```
[3:23▲user@ws▲p1]▲./a.out↵
init▲capacity:▲-1↵
input:▲q↵
↵
size:▲0↵
capacity:▲2↵
content:▲(empty)↵
[3:23▲user@ws▲p1]▲./a.out↵
init▲capacity:▲0↵
input:▲q↵
↵
size:▲0↵
capacity:▲2↵
content:▲(empty)↵
[3:23▲user@ws▲p1]▲./a.out↵
init▲capacity:▲1↵
input:▲1▲q↵
↵
size:▲1↵
capacity:▲1↵
content:▲[1]↵
[3:23▲user@ws▲p1]▲./a.out↵
init▲capacity:▲1↵
input:▲1▲2▲q↵
↵
size:▲2↵
capacity:▲2↵
content:▲[1,▲2]↵
[3:23▲user@ws▲p1]▲./a.out↵
init▲capacity:▲1↵
input:▲1▲2▲3▲4▲5▲6▲7▲8▲9▲10▲q↵
↵
```

```
size:▲10↵
capacity:▲16↵
content:▲[1,▲2,▲3,▲4,▲5,▲6,▲7,▲8,▲9,▲10]↵
[3:23▲user@ws▲p1]
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題將以上述提供的 Makefile 進行編譯，請同學作答時 **“以題目提供的文件為主”**。
- 本題應繳交檔案如下（**define.h**、**main.c**及**Makefile** 則不需繳交）：
  - **numBox.c**
  - **numBox.h**

---

p2 文字內容替換

本題需要請同學實作一個C語言程式，可以將一段輸入的文字內容中的特定內容加以替換。

請參考以下測試檔的範例：

```
burger burger burger
burger
soda
```

在上面的測試檔中：

- 第一行輸入的是原始文字內容
- 第二行是指定所要替換的字串
- 第三行是使用者欲替換的字串

經程式替換後的輸出如下：

```
[3:23▲user@ws▲p2]▲./a.out < testfile_1.in↵
soda▲soda▲soda↵
[3:23▲user@ws▲p2]
```

請參考以下 main.c 的內容：

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

```
#include "replace.h"

int main()
{
    char* str = (char*)calloc(1024, sizeof(char));

    char target[128];
    memset(target, 0, sizeof(target));
    char replace[128];
    memset(replace, 0, sizeof(replace));

    scanf(" %1023[^\n]", str);
    scanf(" %127[^\n]", target);
    scanf(" %127[^\n]", replace);

    strReplace(str, target, replace);
    printf("%s\n", str);

    return 0;
}
```

因此，本題需請同學定義並實作下列函式：

- **strReplace()**

完成後，將內容放入以下檔案：

- **replace.c**：此文件需包含上述 **函式** 的實作。
- **replace.h**：此文件需包含上述 **函式** 的定義。

另外，本題目使用以下 Makefile 進行批改：

```
all: main.c replace.o
    gcc main.c replace.o
replace.o: replace.c
    gcc -c replace.c
clean:
    rm -f *.*~ a.* *.o *.o~
```

本題有提供測試檔，請同學根據下面路徑自行複製：

- /home/stu/public/c2026s/c.hw6/p2/testfile_1.in
- /home/stu/public/c2026s/c.hw6/p2/testfile_2.in
- /home/stu/public/c2026s/c.hw6/p2/testfile_3.in

本題的執行結果可參考如下：

```
[3:23▲user@ws▲p2]▲./a.out < testfile_1.in↵
soda▲soda▲soda↵
[3:23▲user@ws▲p2]▲./a.out < testfile_2.in↵
Yet▲even▲in▲His▲deathless▲state,▲the▲Emperor▲continues▲His▲eternal▲vigilance.▲Mighty▲battlefleets▲cross▲the▲demon-infested▲miasma▲of▲the▲Warp,▲the▲only▲route▲between▲distant▲stars,▲their▲way▲lit▲by▲the▲Astronomican,▲the▲psychic▲manifestation▲of▲the▲Emperor's▲will.▲Vast▲armies▲give▲battle▲in▲His▲name▲on▲uncounted▲worlds.↵
[3:23▲user@ws▲p2]▲./a.out < testfile_3.in↵
In▲Lorem▲ipsum,▲where▲the▲god▲meets▲the▲world,▲and▲Lorem▲ipsum▲is▲also▲Lorem▲ipsum▲of▲the▲world,▲and▲Lorem▲ipsum▲is▲also▲Lorem▲ipsum▲of▲the▲god,▲in▲replacement▲of▲Lorem▲ipsum▲is▲by▲Lorem▲ipsum▲of▲the▲word,▲which▲constrains▲the▲rule▲of▲Lorem▲ipsum,▲to▲be▲holded▲by▲Lorem▲ipsum.↵
[3:23▲user@ws▲p2]
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題將以上述提供的 Makefile 進行編譯，請同學作答時 **“ 以題目提供的文件為主 ”**。
- 本題應繳交檔案如下（**main.c**及**Makefile** 則不需繳交）：
  - **replace.c**
  - **replace.h**

---

p3 通訊錄

在日常生活中我們通常會使用通訊錄來幫我們記錄重要的人的聯絡資料，而這些資料有可能是自己的親朋好友或者是自己上班時的客戶等，

因此，本題需請同學實作一個簡單的C語言通訊錄來記錄這些聯絡資料。首先，請參考以下題目提供的標頭檔 **define.h**。

```
#define GROWTH_RATE 2
#define SHRINK_RATE 0.5

typedef struct
{
    char name[21];
    char phone[11];
    char email[65];
} contact;

typedef struct
{
```

```
    contact* contacts;
    int size;
    int capacity;
} contactBook;
```

在標頭檔中有以下兩個結構體來幫助同學完成本次作業：

- **contact 結構體**:
  - **name**: 聯絡人的姓名，其長度不超過20個字元。
  - **phone**: 聯絡人的電話，其長度不超過10個字元。
  - **email**: 聯絡人的電子郵件，其長度不超過64個字元。
- **contactBook 結構體**:
  - **contacts**: 指向所有目前已記錄聯絡人(contact)陣列的指標。
  - **size**: 紀錄目前存放多少聯絡人。
  - **capacity**: 記錄目前contacts指標指向的陣列實際向系統申請多少的空間(capacity)

在**main.c**中，**contactBookInit()**函式以幫同學完成**contactBook結構體**的記憶體配置與初始化，並請同學透過此結構體實作以下指令：

- **add**: 新增一筆聯絡資料
  - 在增加時，如果增加後的 **大小(size) $\ge$ 容量(capacity)** 時，
    在 **contactBook結構體** 中的 **contacts陣列** 需以 **GROWTH_RATE(請參考上面標頭檔 define.h)** 的倍率增加當前容量。
- **del**: 刪除欲想要的聯絡人
  - 如果找不到要刪除的聯絡人時程式，需輸出“ **Contact not found.** ”並立即返回主程式。
  - 在刪除時，在操作成功後如果 **當前容量(capacity) $\times$ SHRINK_RATE(請參考上面標頭檔 define.h) $\ge$ 當前大小(size)** 時，
    在 **contactBook結構體** 中的 **contacts陣列** 需以 **SHRINK_RATE** 的倍率縮減當前容量。
- **ls**: 列出當前所記錄的聯絡人資訊



實作注意

- 本題假設 **所有的輸入資料皆正確**，因此請同學不用處理輸入邊界問題。

請參考以下 main.c 的內容：

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

#define INIT_CAP 2

#include "contact.h"
```

```
contactBook contactBookInit()
{
    contactBook list;
    list.size = 0;
    list.capacity = INIT_CAP;
    list.contacts = calloc(list.capacity, sizeof(contact));
    return list;
}

int main()
{
    contactBook book = contactBookInit();
    char command[5];

    while (1)
    {
        printf("command (add, del, ls, exit): ");
        scanf(" %[^\n]s", command);

        if (strcmp(command, "add") == 0)
        {
            printf("\n");
            contactBookAdd(&book);
            printf("\n");
        }
        else if (strcmp(command, "del") == 0)
        {
            printf("\n");
            contactBookRemove(&book);
            printf("\n");
        }
        else if (strcmp(command, "ls") == 0)
        {
            printf("\n");
            contactBookList(&book);
            printf("\n");
        }
        else if (strcmp(command, "exit") == 0)
        {
            break;
        }
        else
        {
            printf("\n");
            printf("  Invalid command!\n");
            printf("\n");
        }
    }

    free(book.contacts);
    return 0;
```

```
}
```

因此，本題需請同學定義並實作下列函式：

- **contactBookAdd()**
- **contactBookRemove()**
- **contactBookList()**

完成後，將內容放入以下檔案：

- **contact.c**：此文件需包含上述 **函式** 的實作。
- **contact.h**：此文件需包含上述 **函式** 的定義。

另外，本題目使用以下 Makefile 進行批改：

```
all: main.c contact.o
    gcc main.c contact.o
contact.o: contact.c
    gcc -c contact.c
clean:
    rm -f *.*~ a.* *.o *.o~
```

本題的執行結果可參考如下：

```
[3:23▲user@ws▲p3]▲./a.out↵
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(0),▲capacity(2)↵
▲▲No▲contacts▲to▲display.↵
↵
command▲(add,▲del,▲ls,▲exit):▲del↵
↵
▲▲No▲contacts▲to▲remove.↵
↵
command▲(add,▲del,▲ls,▲exit):▲eixt↵
↵
▲▲Invalid▲command!↵
↵
command▲(add,▲del,▲ls,▲exit):▲exit↵
[3:23▲user@ws▲p3]
```

```
[3:23▲user@ws▲p3]▲./a.out↵
command▲(add,▲del,▲ls,▲exit):▲add↵
```

```
↵
Name:▲Marcus↵
Phone:▲0913245678↵
Email:▲mkx1234@email.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲add↵
↵
Name:▲John↵
Phone:▲0987654321↵
Email:▲jj78901@smail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(2),▲capacity(2)↵
▲▲Name,▲Phone,▲Email↵
▲▲Marcus,▲0913245678,▲mkx1234@email.com↵
▲▲John,▲0987654321,▲jj78901@smail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲add↵
↵
Name:▲Alice↵
Phone:▲0922334455↵
Email:▲alicewashere388@utmail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(3),▲capacity(4)↵
▲▲Name,▲Phone,▲Email↵
▲▲Marcus,▲0913245678,▲mkx1234@email.com↵
▲▲John,▲0987654321,▲jj78901@smail.com↵
▲▲Alice,▲0922334455,▲alicewashere388@utmail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲exit↵
[3:23▲user@ws▲p3]
```

```
[3:23▲user@ws▲p3]▲./a.out↵
command▲(add,▲del,▲ls,▲exit):▲add↵
↵
Name:▲TestUser↵
Phone:▲0123456789↵
Email:▲test@mail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(1),▲capacity(2)↵
▲▲Name,▲Phone,▲Email↵
▲▲TestUser,▲0123456789,▲test@mail.com↵
↵
```

```
command▲(add,▲del,▲ls,▲exit):▲del↵
↵
Name:▲TestUsr↵
▲▲Contact▲not▲found.↵
↵
command▲(add,▲del,▲ls,▲exit):▲del↵
↵
Name:▲TestUser↵
↵
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(0),▲capacity(1)↵
▲▲No▲contacts▲to▲display.↵
↵
command▲(add,▲del,▲ls,▲exit):▲add↵
↵
Name:▲UserA↵
Phone:▲1111111111↵
Email:▲a@mail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲add↵
↵
Name:▲UserB↵
Phone:▲2222222222↵
Email:▲b@mail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲add↵
↵
Name:▲UserC↵
Phone:▲3333333333↵
Email:▲c@mail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(3),▲capacity(4)↵
▲▲Name,▲Phone,▲Email↵
▲▲UserA,▲1111111111,▲a@mail.com↵
▲▲UserB,▲2222222222,▲b@mail.com↵
▲▲UserC,▲3333333333,▲c@mail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲del↵
↵
Name:▲UserD↵
▲▲Contact▲not▲found.↵
↵
command▲(add,▲del,▲ls,▲exit):▲del↵
↵
Name:▲UserB↵
↵
```

```
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(2),▲capacity(2)↵
▲▲Name,▲Phone,▲Email↵
▲▲UserA,▲1111111111,▲a@mail.com↵
▲▲UserC,▲3333333333,▲c@mail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲del↵
↵
Name:▲UserA↵
↵
command▲(add,▲del,▲ls,▲exit):▲ls↵
↵
▲▲size(1),▲capacity(1)↵
▲▲Name,▲Phone,▲Email↵
▲▲UserC,▲3333333333,▲c@mail.com↵
↵
command▲(add,▲del,▲ls,▲exit):▲exit↵
[3:23▲user@ws▲p3]
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題將以上述提供的 Makefile 進行編譯，請同學作答時 **“以題目提供的文件為主”**。
- 本題應繳交檔案如下（**define.h**、**main.c**及**Makefile** 則不需繳交）：
  - **contact.c**
  - **contact.h**

---

p4 五子棋-洞三

承接 作業5(c.hw5)的活三與死四，在實際對弈中，並非所有棋型都是連續排列的。「洞三」（又稱跳三）是一種具有間隔的進攻棋型，雖然中間存在空位，但其威脅性並不亞於標準活三。所謂「洞三」是指由三顆同色棋子與一個中洞組成的四格區間，且該區間的前後兩端都沒有棋子。換句話說，這三顆棋子若為己方，下一手填補中洞即可形成活四。

為簡化規則，本題不需找出活三。

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。

可參考以下 main.c 的內容，讀入棋盤：

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

int main()
```

```
{
char chessboard[19][19];

for(int i=0;i<19;i++)
    for(int j=0;j<19;j++)
        scanf(" %c",&chessboard[i][j]);

isHoleThree(chessboard);
}
```

本題的執行結果可參考如下：

```
[3:23 user@ws p4] ./a.out▲<▲testfile_1.cb↵
Hole▲Three▲found:▲(H,▲▲7)▲(I,▲▲7)▲(K,▲▲7)↵
Hole▲Three▲found:▲(H,▲▲9)▲(J,▲▲9)▲(K,▲▲9)↵
[3:23 user@ws p4] ./a.out▲<▲testfile_2.cb↵
Hole▲Three▲does▲not▲exist.↵
[3:23 user@ws p4]
```

本題測試檔路徑：

- /home/stu/public/c2026s/c.hw6/p4/testfile_1.cb
- /home/stu/public/c2026s/c.hw6/p4/testfile_2.cb



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. 輸出參照 作業5(c.hw5) p2的輸出格式
    1. 若找不到活三則顯示 Hole▲three▲does▲not▲exist.↵



- 本題批改時將採**人工批改**
- 請勿使用**AI撰寫**，若是助教發現疑似使用AI撰寫，將會約談同學。
- 約談過程中，若對本人繳交之程式碼內容回答模糊、邏輯不通，將以**0分**計算。

---

p5 五子棋-洞死四

承上題，修改程式碼增加功能，讓程式能夠找出棋盤中所有洞死四的位置。所謂「洞死四」是指已經由四顆同色棋子與一個中洞組成的五格區間（例如：1 . 1 1 1或1 1 . 1 1），次一手有一點可形成五子連線，若能正確識別，玩家便能提前預判對手的絕殺點。同樣地，本題只需找出連續五個位置中，由四顆同色棋子包含一個中洞的棋型，且該五格區間的頭尾外側已有一端被對手防堵（或是棋盤邊界）。若兩端

皆通，本題亦統一視為偵測目標。

為簡化規則，本題不需找出死四。

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。

可參考以下 main.c 的內容，讀入棋盤：

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

int main()
{
char chessboard[19][19];

for(int i=0;i<19;i++)
    for(int j=0;j<19;j++)
    scanf(" %c",&chessboard[i][j]);

isHoleDeadFour(chessboard);
}
```

本題可參考以下的執行結果：

```
[3:23 user@ws p5] ./a.out ▴<▴ testfile_1.cb↵
Hole▴Dead▴Four▴found:▴(I,▴▴9)▴(J,▴▴9)▴(L,▴▴9)▴(M,▴▴9)↵
Hole▴Dead▴Four▴found:▴(K,▴10)▴(J,▴11)▴(I,▴12)▴(G,▴14)↵
Hole▴Dead▴Four▴found:▴(L,▴10)▴(L,▴11)▴(L,▴12)▴(L,▴14)↵
Hole▴Dead▴Four▴found:▴(O,▴13)▴(P,▴13)▴(Q,▴13)▴(S,▴13)↵
[3:23 user@ws p5] ./a.out ▴<▴ testfile_2.cb↵
Hole▴Dead▴Four▴does▴not▴exist.↵
[3:23 user@ws p5]
```

本題測試檔路徑：

- /home/stu/public/c2026s/c.hw6/p5/testfile_1.cb
- /home/stu/public/c2026s/c.hw6/p5/testfile_2.cb



- 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
- Makefile 會使用到的檔案必須繳交，否則將導致編譯錯誤。
- 輸出參照 作業5(c.hw5) p2的輸出格式
    - 若找不到洞死四則顯示 Hole▴Dead▴Four▴does▴not▴exist.↵



- 本題批改時將採**人工批改**

- 請勿使用**AI撰寫**，若是助教發現疑似使用AI撰寫，將會約談同學。
- 約談過程中，若對本人繳交之代碼內容回答模糊、邏輯不通，將以**0分**計算。

p6 五子棋-參賽

請參考 Gomoku參賽指南，完成第一次的參賽作品上傳。
國立屏東大學資工系電腦五子棋 AI 競賽平台 五子棋競賽網站連結。

- 本題只要 **上傳至參賽平台** 並 **檢查是否能夠由平台順利完成編譯** 即完成本題目，**不需繳交至 turnin**
- 本題將作業截止日期後進行第一次初賽。

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: 343750

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2026spring:hw6

Last update: **2026/05/20 02:21**