2024 NCKU Program Design I Homework 6

> 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))

Homework 6 Information

Deadline: 12/4 23:59:59

> 不接受遲交
>
> Late submissions are not accepted.

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 `HW6` 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 `HW6` directory, you need to create 8 files with name "pA.c", "pA.h", "pB.c", "pB.h", "pC.c", "pC.h", "pD.c" and "pD.h" respectively.
- You can directly use the command `hw6` to check whether the result of each question is correct.

Tasks

Problem A. Let's Try a new style submission (pA.h, pA.c) - 25%

Due to the requirements of future assignments, let's learn how to link two pieces of code together. For this type of problem, we will provide a `main.c` file that contains part of the program. Within this code, some function calls require you to complete their implementation!

Let's use a simple example to demonstrate: suppose the task is to input two numbers and then output the sum of two numbers.

main.c

```
#include <stdio.h>
#include "pA.h"
int main() {
    int a,b;
    scanf("%d %d", &a, &b);
    int ans = add(a,b); // call this function from pA
    printf("%d", ans);
    return 0;
}
```

You need to use `pA.h` to define the `add` function. A `.h` file is a header file, which can be thought of as an interface. Typically, in this type of file, we define functions or certain structures (structs), while the detailed implementation is written in a `.c` file.

If a program wants to use the functions provided by `pA` , it only needs to include `#include "pA.h"` and link the two files during compilation to utilize it.

Since this is a demonstration question, the `pA.h` file is already provided. You only need to write the `pA.c` file.

pA.h

```
int add(int a, int b);
```

pA.c

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

int add(int a, int b) {
    // please write your code here!
    // Hints: return the value of a + b
}
```

Compile Command

```
gcc main.c pA.c -o pA
```

Sample Input 1

```
4 5
```

Sample Output 1

```
9
```

Problem B. Sum of 2D Arrays (pB.h, pB.c) - 25%

In `main.c`, there are two 2D arrays, `A` and `B`. Your task is to design a function called `solve`, which calculates a new 2D array `C` such that:

$$C[i][j] = A[i][j] + B[i][j]$$

Additionally, you need to implement a function called `get_value` to retrieve values from the resulting array `C`.

Function Requirements

Function: `void solve()`

This function will receive the dimensions `n` and `m`, along with references to the arrays `A` and `B`. The function will calculate the result of `C` based on the formula:

$$C[i][j] = A[i][j] + B[i][j]$$

Function: `int get_value()`

This function will take two parameters `x` and `y`, and it must return the value of `C[x][y] = A[x][y] + B[x][y]`.

main.c

```c
#include <stdio.h>
#include "pB.h"

int A[100][100], B[100][100];

void input(int n, int m, int (*arr)[100]) {
    for(int i=0;i<n;i++) {
        for(int j=0;j<m;j++) {
            scanf("%d", &arr[i][j]);
        }
    }
}
int main() {
    int n, m;
    scanf("%d %d", &n, &m);
    input(n, m, A);
    input(n, m, B);

    solve(n, m, A, B);
    for(int i=0;i<n;i++) {
        for(int j=0;j<m;j++) {
            int val = get_value(i, j);
            printf("%d", val);
            if( j != m - 1 ) printf(" ");
        }
        printf("\n");
    }

    return 0;
}
```

Hints (pB.h)

```c
void solve(int n, int m, int (*A)[100], int (*B)[100]);

int get_value(int x, int y);
```

Compile Command

```
gcc main.c pB.c -o pB
```

Sample Input 1

```
3 3
1 1 1
1 1 1
1 1 1
2 2 2
2 2 2
2 2 2
```

Sample Output 1

```
3 3 3
3 3 3
3 3 3
```

Problem C. Fibonacci Sequence (pC.h, pC.c) - 25%

Fibonacci Sequence: (f(n) = f(n-1) + f(n-2))

Function Requirements

Function: `void f()`

This function will take a positive integer `n` as input. You are required to modify the value of the variable `ans` , and update it to `f(n)` using a pointer.

- $1 \leq n \leq 8$

main.c

```
#include <stdio.h>
#include "pC.h"
int main() {
    int n;
    scanf("%d", &n);
    int ans = 0;
    f(n, &ans);
    printf("%d", ans);

    return 0;
}
```

Problem D. Swap a 3D-Array ! (pD.h, pD.c) - 25%

Function Requirements

Function: `void exchange()`

Given references to two three-dimensional arrays, swap these two three-dimensional arrays.

Function: `void modify()`

Given an identifier `id` and three indices `x` , `y` , `z` along with a positive integer `value` , modify the element at index `[x][y][z]` of the `id` -th three-dimensional array to `value` . Additionally, update the value of `id` (passed by reference) to the result of $x + y + z$.

Function: `int get_value()`

Given a reference to `arr` and the parameters `id` , `x` , `y` , `z` , return the value of $arr[id][x][y][z]$.

- $1 \leq q \leq 1000$
- $1 \leq opt \leq 3$
- $0 \leq id \leq 9$
- $0 \leq x, y, z \leq 19$
- $0 \leq id1, id2 \leq 9$
- $1 \leq val \leq 10^9$

main.c

```c
#include <stdio.h>
#include "pD.h"

int arr[10][20][20][20]; // 10 個 3-D 陣列
int main() {
        int q;
        scanf("%d", &q); // 總操作次數
        while(q--) {
                int opt;
                scanf("%d", &opt);
                if( opt == 1 ) { // 操作 1
                        int id, x, y, z, val;
                        scanf("%d %d %d %d %d", &id, &x, &y, &z, &val);
                        modify(arr[id], &id, x, y, z, val);
                        printf("%d\n", id);
                }
                else if( opt == 2 ) { // 操作 2
                        int id1, id2;
                        scanf("%d %d", &id1, &id2);
                        exchange(arr[id1], arr[id2]);
                }
                else if( opt == 3 ) { // 操作 3
                        int id, x, y, z;
                        scanf("%d %d %d %d", &id, &x, &y, &z);
                        int ans = get_value(arr, id, x, y, z);
                        printf("%d\n", ans);
                }
        }

        return 0;
}
```

Sample Input 1

```
9
1
5 10 10 10 100
3
5 10 10 9
3
5 10 10 10
1
7 4 5 6 2024
2
7 5
3
5 10 10 10
3
7 4 5 6
3
7 10 10 10
3
5 4 5 6
```

Sample Output 1

```
30
0
100
15
0
0
100
2024
```