程式設計(一) - 作業 7

NCKU Program Design I Homework 7

The key focus of this Assignment

- Recursion
- array

Before Start

- Don't attack any system otherwise you will fail this course.

- 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.

DeadLine: 11/16 00:00

- No Delay Submission ！ ！ ！

Submission

- 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 1 files with name “hw7.c”
- You need to compile your program by yourself, and create 1 executable files with the filenames “hw7”

Kirby is a System Administrator

- 卡比是 程式設計(一) 的系統管理工程師
- Kirby is a system administrator

Kirby is a System Administrator

- 我們的主機最近遇到了一些問題，有些同學的檔案太大，把我們的硬碟塞爆了
- 因此卡比必須設計一個程式來管理硬碟
- We've been experiencing some issues with our server recently, as some students have been uploading very large files, filling up our hard drive
- So Kirby needs to design a program to manage the hard drive.

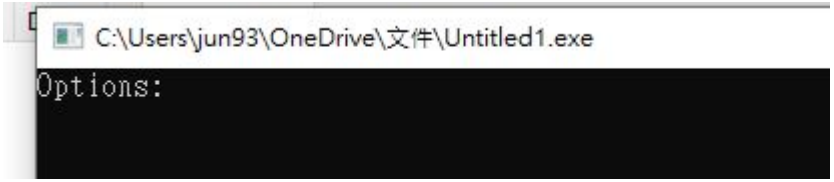
Kirby is a System Administrator

- 目前已知硬碟裡面的檔案有以下規定：
  - 硬碟裡面不能超過 20 個檔案 (可以是 20 個)
  - 一旦超過 20 個檔案，就必須刪除 k 個檔案 (k 是使用者輸入)
- The hard drive cannot exceed 20 files (can be exactly 20 files). Once it exceeds 20 files, it must delete k files (where k is user input)

Kirby is a System Administrator

- 硬碟管理程式需要具備三種功能：
  - 1. 新增檔案
  - 2. 查詢檔案
  - 3. 結束程式
- The disk management program needs to have two functions:
  - 1. Add some files.
  - 2. Query the file
  - 3. Exit the process.

General



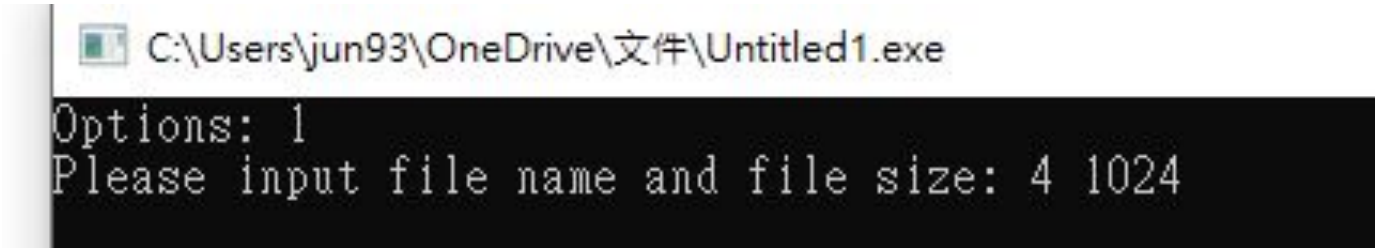
- 使用者在決定要使用什麼功能的時候，必須輸出 "Options: "
- 並等待使用者輸入
- When the user decides which function to use, they must output 'Options:' and wait for user input.
- 每一次使用者執行完某一個操作時，都必須讓使用者重新回到選擇操作的功能
- After each time the user completes an operation, they must be returned to the option selection.

Options 1

- 如果使用者輸入 1 表示執行新增檔案的功能
- 接下來輸出 "Please input file name and file size: "
- 輸出後，讓使用者輸入兩個正整數，第一個表示檔案名稱，第二個數字表示這一個檔案的大小
- "If the user inputs 1, it indicates the execution of the 'Add a file' function. Next, output 'Please input file name and file size:' and allow the user to input two positive integers, where the first represents the file name and the second represents the size of the file.

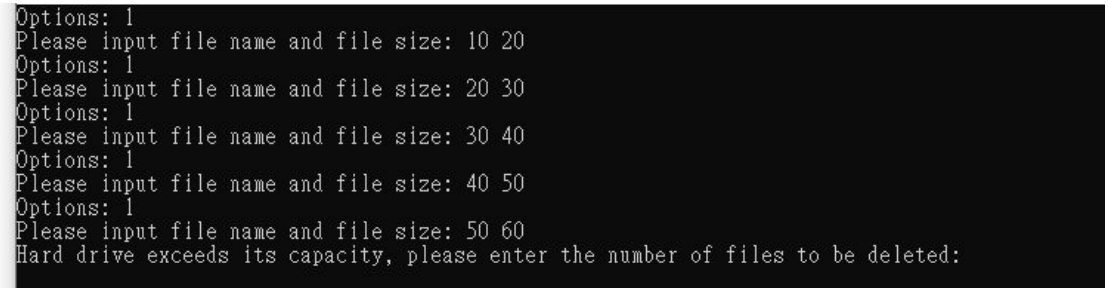
Options 1

- file name: 4
- size of file: 1024



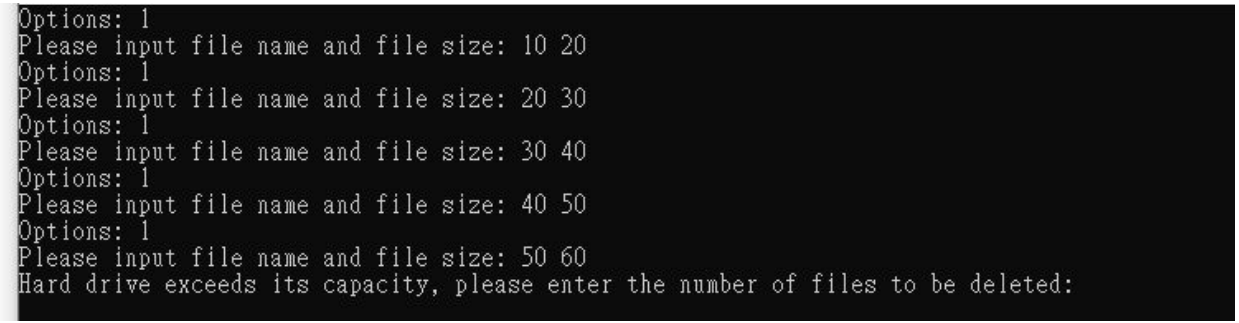
Options 1

- 輸入完後必須把這一個檔案新增到硬碟
- 新增完後如果硬碟檔案超過 20 個，必須輸出
- "Hard drive exceeds its capacity, please enter the number of files to be deleted: "
- 並讓使用者輸入兩個數字 k 跟 m



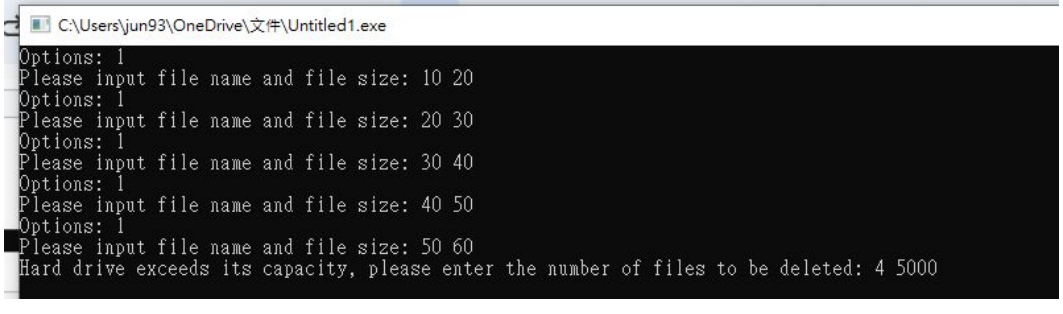
Options 1

- After entering, the file must be added to the hard drive. After adding, if the hard drive files exceed 20, it must output 'Hard drive exceeds its capacity, please enter the number of files to be deleted:' and allow the user to input two numbers, k and m.



Options 1

- k 表示要刪除 k 個檔案，m 表示要刪除的大小總和
- 假設輸入 4 5000
  - 表示必須刪除 4 個檔案
  - 這 4 個檔案的大小總和必須要越接近 5000 越好
  - 你必須輸出你刪除的檔案名稱，我們的腳本會去檢查你刪除的檔案是否是最佳的選擇



Options 1

- k represents the number of files to be deleted, and m represents the total size to be deleted. For example, if you input 4 5000, it means you must delete 4 files, and the total size of these 4 files should be as close to 5000 as possible. You must output the names of the files you delete, and our script will check whether the files you delete are the optimal choices.

Options 1

- 你必須輸出你刪除的**檔**案名稱，(總共 k 個)
- 任意兩個**檔**案名稱之間有一個空格，最後換行

```
Options: 1
Please input file name and file size: 2023 4000
Options: 1
Please input file name and file size: 145 3000
Options: 1
Please input file name and file size: 1 100000000
Options: 1
Please input file name and file size: 2 2000
Options: 1
Please input file name and file size: 80 80000000
Hard drive exceeds its capacity, please enter the number of files to be deleted: 2 6500
2023 145
Options: 1
Please input file name and file size: 10 20
```

Options 1

- You must output the names of the files you delete (total of k files). There should be one space between any two file names, and end with a newline.

```
Options: 1
Please input file name and file size: 2023 4000
Options: 1
Please input file name and file size: 145 3000
Options: 1
Please input file name and file size: 1 100000000
Options: 1
Please input file name and file size: 2 2000
Options: 1
Please input file name and file size: 80 80000000
Hard drive exceeds its capacity, please enter the number of files to be deleted: 2 6500
2023 145
Options: 1
Please input file name and file size: 10 20
```

Options 1

- 最佳刪除的方案可能有很多種
- 任意輸出一種最佳方案我們都會算你對
- 我們的腳本會針對你的方案去驗證你的輸出是否正確
- There can be multiple optimal deletion strategies, and we will consider your output correct for any valid optimal strategy. Our script will validate your output based on the strategy you choose.

Options 1

- 舉例來說，如果 k = 2，m = 6500
- 如果有某三個檔案的名字與大小分別是：
  - { 2023 , 4000 }
  - { 2 , 2000 }
  - { 145 , 3000 }
- 你輸出 2023 2 或 2023 145 或 2 2023 或 145 2023 其中一個都可以，只要是最佳的選法皆可

Options 1

- For example, if k = 2 and m = 6500, and there are three files with names and sizes as follows:
  - { 2023 , 4000 }
  - { 2 , 2000 }
  - { 145 , 3000 }
- You can output either '2023 2' or '2023 145' or '2 2023' or '145 2023'—any of these is acceptable as long as it is one of the optimal choices.

Options 1

- 輸出完 k 個要刪除的檔案後，那 k 個檔案就已經不在硬碟裡面了
- After outputting the names of the k files to be deleted, those k files will no longer be on the hard drive.
- 特別注意在刪除的時候，剛新增的那筆檔案也要考慮進去，也就是說你必須從 21 個檔案中選出 k 個刪除
- Please pay special attention that when deleting, the recently added file should also be considered. In other words, you must select k files for deletion from among the 21 files.

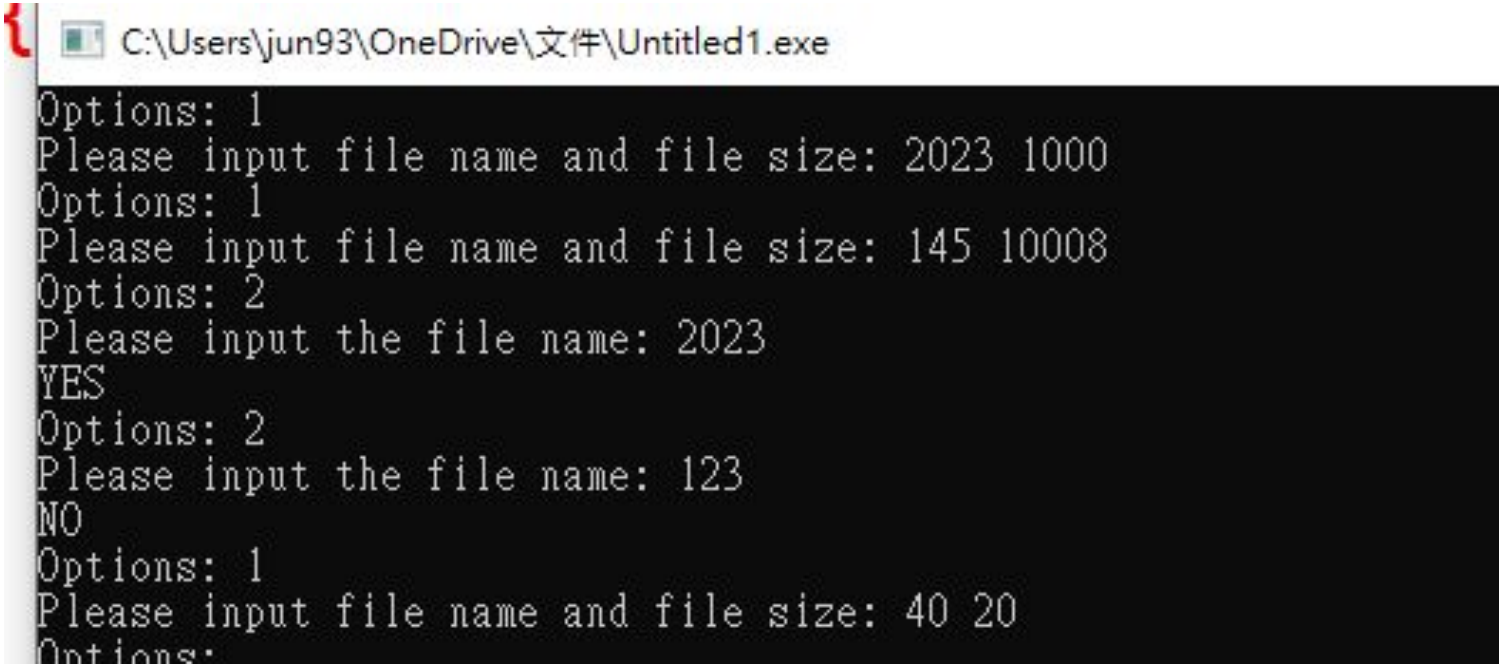
Options 2

- 操作 2 一開始必須先輸出“Please input the file name: ”
- 並等待使用者輸入檔案名稱
- 輸入完檔案名稱之後，你的程式必須回報硬碟中是否存在此檔案
- 如果有則輸出 "YES"，反之則輸出 "NO"

Options 2

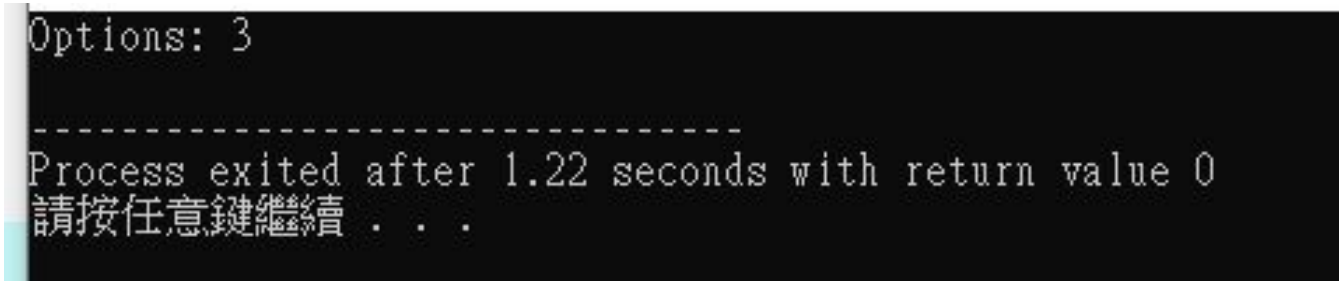
- Operation 2 must start by outputting 'Please input the file name:' and waiting for the user to input the file name. After the user enters the file name, your program must report whether this file exists on the hard drive. If it exists, output 'YES'; otherwise, output 'NO'.

Options 2


C:\Users\jun93\OneDrive\文件\Untitled1.exe
Options: 1
Please input file name and file size: 2023 1000
Options: 1
Please input file name and file size: 145 10008
Options: 2
Please input the file name: 2023
YES
Options: 2
Please input the file name: 123
NO
Options: 1
Please input file name and file size: 40 20
Options:

Options 3

- 關閉程式
- End of process.



Test Data Limits

- 保證不會亂輸入東西
- k <= 21 , m <= 10^9
- 檔案名稱跟大小保證是正整數 且不超過 10^9
- 操作 2 查詢的檔案名稱保證是正整數且不超過 10^9
- The file names and sizes are guaranteed to be positive integers and not exceed 10^9. The file name queried in operation 2 is guaranteed to be a positive integer and not exceed 10^9.

Test Data Limits

- 最多只會執行 100 次操作
- There will be a maximum of 100 operations executed.
- 保證測試資料不會讓硬碟內有兩個以上相同名稱的檔案
- Ensure that the test data does not contain more than one file with the same name on the hard drive.

Subtask & Grading (100 %)

- Subtask 1 (10 %)
  - 你的程式在一開始就輸入 3 可以直接關起來
  - You can input 3 at the beginning to close the program immediately.
- Subtask 2 (30 %)
  - 你的程式可以判斷檔案是否存在
  - 測試資料保證檔案不會超過 20 個
  - Your program can determine whether a file exists.
  - The test data guarantees that there will not be more than 20 files

Subtask & Grading (100 %)

- Subtask 3 (30 %)
  - 你的程式可以依照要求刪除正確數量的檔案
  - 但你的方法不是最佳解
  - Your program can delete the correct number of files as required, but your method is not the optimal solution.
- Subtask 4 (30 %)
  - 你的程式完全正確
  - Your program is entirely correct.

Hints

- 這個作業，適時的運用陣列或變數紀錄資訊是一件很重要的事情
- 找出最佳解的方式可以透過遞迴去求得
- 刪除時每一個檔案只有兩種選擇：
  - 要刪除這個檔案
  - 不要刪除這個檔案
- 透過遞迴的方式嘗試每一種方案，選出最好的那一個

Hints

- In this assignment, using arrays or variables to record information appropriately is crucial.
- Finding the optimal solution can be achieved through recursion. When deleting, each file has only two choices: to delete the file or not to delete it. By recursively trying every possible scenario, you can determine the best one.

Hints

```
bool choose[22]; // 利用 choose[i] 紀錄第 i 個檔案是否有被刪除

void solve(int idx,int sum) // 目前正在考慮第 idx 個檔案，且目前所有已經選擇要刪除的檔案總大小是 sum
{
    // 遞迴終止條件（已經選了 k 個 ? , 已經沒得選了，但選的東西沒有超過 k 個 ? ）

    choose[idx] = true;
    solve(idx+1,sum+file_size[idx]); // 刪除這個檔案

    //?（助教課有說過的遞迴細節，後來決定不把 idx 刪除時，原本標記的東西必須要 ... ）

    solve(idx+1,sum); // 不要刪除這個檔案
}
```