

第1次練習-練習-PC1

學號：112111234

姓名：阮陳家興

作業撰寫時間：180 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2023/09/22

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

說明程式與內容

開始寫說明，該說明需說明想法，並於之後再對上述想法的每一部分將程式進一步進行展現，若需引用程式區則使用下面方法，若為.cs檔內程式除了於敘述中需註明檔案名稱外，還需使用語法```語言種類 程式碼```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp，下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){  
    ...  
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼```，下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<meta http-equiv="Content-Type" ...>  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            </div>  
    </form>  
</body>  
</html>
```

更多markdown方法可參閱<https://ithelp.ithome.com.tw/articles/10203758>

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

1. 請執行下面Git操作 (請參照題目pdf)

Ans: 完成

2. 請整理課堂上所提的Git指令，顯示其語法語給予明顯的例子

Ans: 1. `git clone (git上的ssh)`

- 目的: 從遠端 Git 倉庫下載整個專案到本地
- 範例: `git clone git@github.com:hag/repository.git`

2. `git add (檔名)` 或 `git add .`

- 目的: 將檔案或所有更改加入暫存區。
- 範例：`git add file.txt` # 加入特定檔案
`git add .` # 加入所有更改

3. `git commit`

- 目的: 將暫存區的更改提交到本地倉庫，通常會附上註解。
- 範例：`git commit "描述所做的更改"`

4. `git push`

- 目的: 將本地倉庫的更改推送到遠端倉庫。
- 範例：`git push origin main` # 推送至 main 分支

5. `git fetch`

- 目的: 從遠端倉庫下載所有新的資料，但不會自動合併到本地分支。
- 範例：`git fetch origin`

6. `git merge (分支)`

- 目的: 合併指定的分支到當前分支。
- 範例：`git merge testdev` # 將 testdev 分支合併到當前分支

7. `git pull`

- 目的: 從遠端倉庫下載資料並自動合併到當前分支，實際上是 `git fetch` 和 `git merge` 的組合。
- 範例：`git pull origin main`

8. `git branch (新的分支名稱)`

- 目的: 創建一個新的分支。
- 範例：`git branch feature/new-feature`

9. `git checkout (分支名稱)`

- 目的: 切換到指定的分支。
- 範例：`git checkout main` # 切換到 main 分支

3. 請先寫出同學認為資料結構一門課每周需要花多少時間？並提出理由。並將「請問要學習大學的資料結構每一週需要對於一般大學生需要多少時間」貼入chatgpt，並將其所建議的敘述貼出後，比較同學所認

為若是每周低於chatgpt所評估的時間，兩者主要差異為何？應該怎麼安排自己時間了解資料結構？

根據我個人的想法，每週要學好資料結構，學生需要大約 3 到 4 小時。這段時間可以用來學習理論，做作業，並實踐寫程式。因為這門課有很多抽象的概念，例如鏈結串列、樹、圖、排序演算法和搜尋演算法等，如果只是學理論而不實踐的話，會很難理解並且容易忘記。

當我詢問ChatGPT關於學習資料結構的時間時，它建議每週 8 到 12 小時。我認為，和我計算的時間相比，ChatGPT的估計稍微高了一些。不過，我覺得如果每週花的時間少於ChatGPT所說的8小時，會很難跟上課程進度並有效地完成作業，因為除了閱讀之外，還需要大量實踐寫程式。

然而，一個普通的學生無法只為了學一門課而花太多時間。因此，我認為為了理解課程並完成作業，我會儘量安排每週至少 6 小時 來學習這門課。可以將時間分配為，週初花2小時來閱讀理論，並在週末花4小時來完成作業並解決一些困難的問題。如果有空閒時間，我會回顧課程內容或尋找更多線上資源來加強我的知識。