Homework #1

Purple Correction Date: 2023/03/12 00:00

Due Time: 2023/03/19 (Sun.) 22:00

Contact TAs: `vegetable@csie.ntu.edu.tw`

Instructions and Announcements

- **NO LATE SUBMISSION OR PLAGIARISM IS ALLOWED.**
- Discussions with others are encouraged. However, you should write down your solutions **in your own words**. In addition, for **each and every** problem you have to specify the references (the URL of the web page you consulted or the people you discussed with) on the first page of your solution to that problem.
- Some problems below may not have standard solutions. We will give you the points if your answer is followed by reasonable explanations.

Submission

- Please place your answers in the same order as the problem sheet and do not repeat problem descriptions, just organize them by problem number in a tidy manner.
- Please zip all the files, including one PDF and two shell scripts, name the zip file "`{your_student_id}.zip`", and submit it through NTU COOL. The zip file should not contain any other files, and the directory layout should be the same as listed below:

```
{your_student_id}/
+-- {your_student_id}.pdf
+-- {your_student_id}_SA/
    +-- p1.sh
    +-- p2.sh
```

Grading

- NA accounts for 50 points while SA accounts for 50 points. The final score is the sum between them.
- It's possible you don't get full credits even if you have the correct answer. You should show how you get the answers step by step and list the references.
- Tidiness score: 3 bonus points, graded by TA.
- Final score = NA score + SA score + tidiness score.

Network Administration

Wireshark/tcpdump

嚴格禁止 DDoS 我們的服務！違者將會遭受到嚴重的懲罰！

Wireshark 和 tcpdump 都是功能強大的網路封包分析工具，其中 Wireshark 是圖形化介面，而 tcpdump 是終端機介面。在這份作業裡，你需要練習並熟悉這兩個工具。

以下題目請詳細寫出過程，只寫答案不會給分喔！

看個影集也會不小心洩漏密碼 ?! (18%)

以下題目請使用系上網路作答，可直接來系館寫作業或使用 VPN 連系上網路（使用 VPN 連線時記得 wireshark 要選對 interface 喔 !）

請執行以下步驟：

- 用 Wireshark 或 tcpdump 開始擷取網路封包。
- 打開瀏覽器，前往 netflix-http。
- 帳號輸入什麼都可以，密碼請輸入你的學號。(不要真的打你的密碼！)
- 目前網站只能在系館連到，或者也可使用系上 VPN。

然後，回答下列問題:

1. 找到那個含有你帳號密碼的封包並附上螢幕截圖。(8%)
2. 重複上列步驟，但這次請前往 netflix-https。請找到那個含有你帳號密碼的封包並附上螢幕截圖，如果找不到的話請說明原因。(10%)

爆炸好快樂 (8%)

請回答以下問題，並詳細說明你是如何得到答案的，若你找不到答案，仍然可以寫下你的想法，我們會視情況給予部份分數。

「叮咚 !」納莎打開門後發現門外的花園裡多了好多包裹，這時他看到地上有一封信，信上寫著：「嗨納莎好久不見，今天是你的生日所以我要送禮物給你，但為了懲罰你之前一次就通過 P 老師的課，所以我寄了一堆包裹給你，但是只有一個是最重要的包裹，其他包裹若是打開的話你房子會被炸掉喔 !」原來包裹是計程修了三次才過的露莎寄來的，納莎瞬間緊張起來，快幫幫納莎找到他的生日禮物，不然他的家就要被炸了！

newflag.pcapng 裡面有一堆包裹，哪個是納莎的生日禮物呢？

Hint: 最重要的包裹！禮物是一個 Flag 開頭的東西

Hint2：打開包裹之後的那個東西裡面應該還隱藏一個檔案，可以找找看用什麼方法可以看到隱藏的檔案

好玩遊戲每天要玩 (8%)

桐人，16 歲，是個劍士，生活在 SAO 74 層，每天靠著狩獵雜燴兔維生，偶爾會到迷宮區的深處對著艾恩葛朗特解放軍軍團的墓塚星爆，以弔念死去的好基友閃耀魔眼。今天桐人也一如往常來到

BOSS 房，當他喊出招式名稱「Starburst Stream」時，他的動作卻不斷回溯，原來是遊戲伺服器又被 DDoS 了。這讓他不斷重複發出「mumumu...」的聲音持續好幾個小時。桐人對於自己被做出社恐行為感到非常不滿，於是他向遊戲開發商 ARGUS 回報了此事，希望能找出兇手並且狠狠的星爆他一頓。

另一方面，身為系統管理員的茅場為了找出兇手是誰，每天都開著古拉牌封包紀錄器 wireshark，bocchi.pcapng是部分記錄下來的封包，已知兇手在這些封包裡面藏了一句話，你能幫助茅場找到這句話，進而找出是誰 DDoS 伺服器嗎？(答案格式：`FLAG{[A-Za-z0-9_]*}`)

Hint：DNS data exfiltration

這麼多的網路協定要是能全部都認識的話該有多好 (16%)

請用 Wireshark 或 tcpdump 擷取下列的封包。在以下的每題當中，請附上封包的螢幕截圖、簡答該協定的用途，並說明該協定運作在 TCP/IP 五層網路架構中的哪些層。

1. ICMP request & reply (2%)
2. DNS query & response (2%)
3. ARP request & reply (4%)
4. DHCP discover & Offer & Request & Ack (8%)

System Administration

UNIX Basics (20%)

I. Basic Permissions (10%)

以下為目錄 `MyAwesomeDirectory` 下的檔案結構及個別權限，其中 `dir*` 為目錄、`file*` 為一般檔案、`link*` 為符號連結（symbolic link）。所有目錄、檔案和符號連結的擁有者都是自己。

```
$ sudo tree -pfi
[dr-xr-xr-x]  .
[d--x--x--x]  ./dir1
[dr--r--r--]  ./dir1/dir1A
[-rw-rw-rw-]  ./dir1/dir1A/file1
[drwxrwxrwx]  ./dir1/dir1B
[-r--r--r--]  ./dir1/dir1B/file2
[drwxrwxrwx]  ./dir2
[-rw-rw-rw-]  ./dir2/file3
[lrwxrwxrwx]  ./dir2/link1 -> ../dir1/dir1B/file2
[drwxrwxrwx]  ./dir3
[lrwxrwxrwx]  ./dir3/link2 -> ../dir1/dir1B/file2
```

回答是否可成功在資料夾 `MyAwesomeDirectory` 下進行以下每小題的動作。除了須回答 true 或 false 外，也須簡單說明理由。若無理由或理由錯誤則該小題不給分。
提示：如果想自己實驗看看指令會不會成功，可以用 'alias <cmd>' 確定指令有沒有被加上其他選項。有些指令在多加其他選項時會需要更多的權限，而本題都沒有加額外選項。

1. `ls dir1`
2. `ls dir1/dir1A`
3. `cd dir1`
4. `cd dir1/dir1A`
5. `echo 'howdy' > dir1/dir1A/file1`
6. `echo 'howdy' > dir1/dir1B/file2`
7. `echo 'howdy' > dir2/file3`
8. `echo 'howdy' > dir2/link1`
9. `rm dir1/dir1B/file2`
10. `rm dir3/link2`

II. ACL (3%)

請先進入工作站 linux7 ，在 `/tmp2/NASA-HW1/` 下建立自己學號的資料夾，英文字母部分需使用小寫，下面題敘會以 `<student_ID>` 代稱。以下每個小題若須使用指令操作，除了須在 linux7 上設定外，也須在 `{your_student_id}.pdf` 寫下設定的指令，若無則對應的小題不給分。

請在資料夾 `<student_ID>` 下新增一個資料夾 `chatroom` 來模擬聊天室，並設定成只有你與你的好朋友能進入。(3%)

1. 請任意選擇一個存在的工作站帳號當作好朋友來設定下面的題目，建議使用非 group `ta` 的帳號，並且在 `{your_student_id}.pdf` 中說明。(0%，但沒說明就無法取得本大題的分數)

2. 設定相關權限，只有自己與好朋友可以檢視資料夾 `chatroom` 的內容 (`ls` 時不會有權限不足的問題) ，以及新增與刪除檔案。(1%)
   提示: 此題可能會需要設定資料夾 `<student_ID>`。

3. 這個聊天室的設計為一個檔案代表一個主題，因此你與你的好朋友們應該都要可以編輯資料夾 `chatroom` 下的所有檔案，才能讓雙方針對同一個主題討論。請設定資料夾 `chatroom` ，使之後新增的檔案都可以被雙方編輯，不需一個一個檔案設定。(1%)

4. 過了一段時間，你發現你們只會在其中一個主題下聊天，因此想將新增主題的功能暫時關閉，也就是將你與好朋友對於資料夾 `chatroom` 的權限設為只能檢視內容，但目前已有的檔案仍可編輯。請對資料夾 `chatroom` 使用 ACL 的 mask 來設定。(1%)

III. Linux / Unix 雜項 (7%)

1. 在 Linux 上你可以用 `chsh` 更改自己預設的 shell。雖然這個指令會變動只有 root 有寫入權限的 `/etc/passwd` ，但執行 `chsh` 卻不用 root 權限。這是因為 `chsh` 有 SUID。請解釋 SUID 的功能。(1%)

2. CSIE Workstation 的 `/tmp2` 所有人都有讀寫權限，那為什麼我們無法刪掉其他人在 `/tmp2` 下的檔案呢？(1%)
   提示：到工作站上看看 `/tmp2` 開的權限是什麼樣子吧

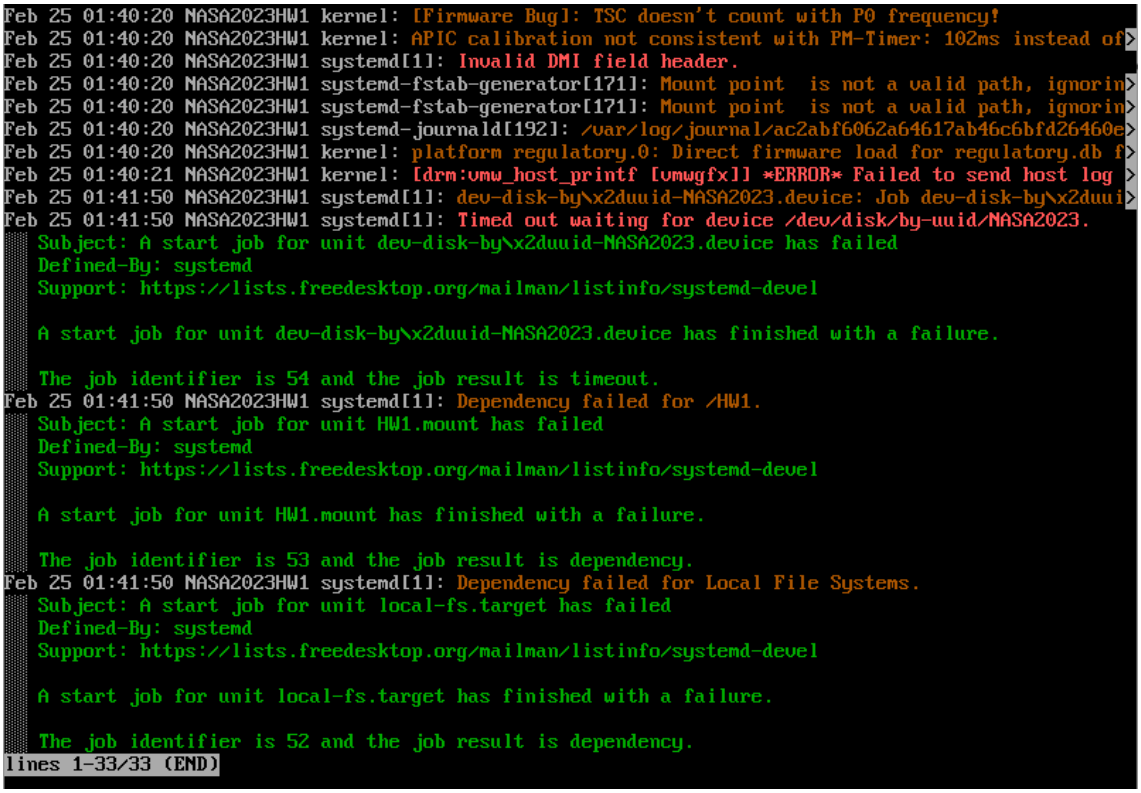
3. 目前 CSIE Workstation 用的是 Arch Linux，而許多其他 SA 維護的服務則是使用 Debian 。請列出一個 Arch Linux 和 Debian 差異。(1%)

4. 在英文的網路論壇上常常可以看到這段 copypasta

   > I'd just like to interject for a moment. What you're referring to as Linux, is in fact, GNU/Linux, or as I've recently taken to calling it, GNU plus Linux. ...

   許多的 Linux 發行版(distro)包含了 GNU 計畫下的軟體套件(例如 GNU coreutils)，不過也有一些發行版是預設使用其他的方案。請舉出一個**預設不是使用 GNU coreutils** 的發行版。(1%)

5. 在大部分的 Linux 系統上，可以使用 `journalctl` 指令來查看系統的各種日誌 (log) 。請使用 `journalctl` 列出**上次開機後**，所有 **warning 等級以上**的記錄，同時**開啟額外的解釋訊息**，並將你用的指令寫在 report 中。(1%)
   提示一：`man journalctl` 看看有哪些選項可以使用吧
   提示二：如果你不太確定自己的指令對不對，可以在這個 VM (使用者 `student` 密碼 `hw1`) 裡面跑看看，應該會看到下圖的輸出



6. `~/.bashrc` 和 `~/.bash_profile` 都是用於 bash 初始化的檔案。請問這兩個檔案各會在什麼樣的情況下被讀取並執行？(1%)

7. NASA 作業很多時候都會提供 VM，而 VM 中可能沒辦法直接使用你本地的 IDE，因此需要使用別的工具來撰寫程式，其中一個選項就是使用 `vim`。為了讓 `vim` 更好用，請設定 `~/.vimrc` 完成以下功能。(1%)

   (a) 顯示每一行的行號，最開頭的那行行號是 1。

   (b) 當前所編輯那行會有 highlight。

   (c) 自動對齊縮排：如果上一行有 2 個 tab 的寬度，按 enter 繼續編輯下一行時會自動保留 2 個 tab 鍵的寬度，依此類推。

   (d) 可以使用游標改變當前位置或是選取文字。

   (e) 當按下 tab 鍵時不會插入 1 個 tab 字元，而是會插入 4 個空白字元（Space）。

   如果你在 SA III P5 的 VM 中完成以上設定，應該可以看到與下圖類似的畫面（雖然能用圖片表達的只有前兩個要求）。

只要針對每一項要求分別寫出 `~/.vimrc` 內相關的程式碼，並附上簡單說明（例如這些參數是為何這樣設定）即可，不需附上無關的程式碼或是截圖。

Shell Scripting (30%)

注意事項

1. 請在 script 前加上 *shebang* 並開啟 script 的執行權限 (`chmod +x`)。
2. 你的 script 必須能在以下幾種 shell 中之一執行，`bash`、`sh`、`zsh`、`fish`、`tcsh` 或 `ksh`。
3. 我們會在工作站測試你的 script，請確認所有使用到的指令在工作站上皆可以正常運作。
4. 此部分需全部使用 shell script 完成，不可在中途執行自己的 python 腳本或使用其他程式語言，如果不確定是否可以使用某些指令，請寄信詢問助教。
5. 請勿在 shell script 內新增檔案，也請勿夾雜惡意程式碼，並在一分鐘內成功執行。

I. RSA, a sibling of NASA? (25 %)

蛤！RSA？那是什麼東西啊。可以吃嗎？身為 NASA 修課學生的你，除了對各種系統指令必須相當熟悉外，也必須得要學會一些酷酷的密碼學知識 (?) 來保障你的資料安全，不然各種外國人會前來嘗試偷取你的資料，讓系上的各種服務陷入危險之中。而在眾多密碼學的東西中，RSA是其中被應用最廣泛的公鑰密碼學演算法之一，在這份作業中，你要嘗試使用 shell script 實作 RSA 演算法。

1. Argument Parser (5%)

在 RSA 演算法中，主要分成產生鑰匙對、加密以及解密三個部分。在這個小題中，你需要分別對三種不同模式檢查參數是否有誤，請依照下列內容讀取相對應的參數，不會有同一個參數重複出現的情況。

```
This command is an RSA algorithm implementation written in shell script.
There are three modes to choose from:

I. Key generation:
usage: ./p1.sh --key-generation <1st prime number> <2nd prime number>
eg: ./p1.sh --key-generation 707981 906313

II. Encrypt mode:
usage: ./p1.sh --encrypt --public-exponent <e> --modulus <n> <file>
eg: ./p1.sh --encrypt --public-exponent 65537 --modulus 641652384053
testfile
usage: ./p1.sh --key-generation <1st prime number> <2nd prime number>
--encrypt <file>
eg: ./p1.sh --key-generation 707981 906313 --encrypt testfile

III. Decrypt mode:
usage: ./p1.sh --decrypt --private-exponent <d> --modulus <n> <file>
eg: ./p1.sh --decrypt --private-exponent 64657547393 --modulus 641652384053
testfile
```

指令內容

1. `-h --help` 輸出使用說明
2. `--key-generation` 會接著讀取兩個質數，並接著產生出可用於 RSA 加密與解密的鑰匙對
3. `--encrypt` 代表使用加密模式
4. `--decrypt` 代表使用解密模式
5. 加密模式時，需要額外提供 `--public-exponent` 以及 `--modulus` 兩個參數，或使用 `--key-generation` 參數產生鑰匙對
6. 解密模式時，需要額外提供 `--private-exponent` 以及 `--modulus` 兩個參數
7. `<file>` 參數代表需要進行加密或解密的檔案，需要確認該檔案為普通文件 (regular file)

補充事項

1. 測資中 `<file>` 不會以 "-" 開頭
2. 測資中輸入的數字可能包含其他字元而導致不合法
3. 給定的數字輸入可能為十六進位表示：以 "0x" 開頭，包含數字 0 至 9 及字母 A 到 F，使用的字母應為全大寫或全小寫
4. 若有任何參數傳遞錯誤，除了印出使用說明外，同時需將 shell script 以 1 作為回傳值結束執行，其餘狀況則以 0 作為回傳值

5. 你可能會發現，shell 內建的計算函數沒有辦法處理大數的問題，或許 bc 是一個好用的東西 (?
6. 數字冪次太大怎麼辦
7. 測資有多行輸入，每個輸入的數字皆保證會用換行符號隔開
8. 範例測資有做更新請大家重新查看

範例合法參數

```
./p1.sh --key-generation 2477 0xb23
./p1.sh --encrypt --public-exponent 65537 --modulus 0x9565766135 testfile
./p1.sh --encrypt --key-generation 0x79FF 29327 testfile
./p1.sh --decrypt --private-exponent 4352849 --modulus 8523649 testfile
```

範例非法參數

```
# no enough parameters
./p1.sh --key-generation
# invalid number
./p1.sh --encrypt --public-exponent 34x7b --modulus ab92o testfile
# invalid hexdemical representation
./p1.sh --encrypt --key-generation 0x79Ff 29327 testfile
# invalid flag -k
./p1.sh --encrypt -k 0x79FF 29327 testfile
# invalid file (testdir is a directory)
./p1.sh --encrypt -k 0x79FF 29327 testdir
```

2. Algorithm implementation (20%)

請延續 1. 寫的 `p1.sh` 並根據參數中所要求的模式進行運算，並將結果輸出。

鑰匙對生成 (5%)

RSA 演算法會根據兩個質數 p, q，用以下的步驟生成鑰匙對（Keys）：

1. 計算 $N = pq$。
2. 計算 $r = \varphi(N)$，φ 是歐拉函數。根據歐拉函數的性質，$\varphi(N) = \varphi(p) \times \varphi(q) = (p-1) \times (q-1)$。
3. 選擇 Public exponent e，此時需要滿足 e 和 r 互質。
4. 使用輾轉相除法求 Private exponent d 使得 $ed \equiv 1 \pmod{r}$，

以下 pseudo code 會根據參數 p 和 q 計算 Public exponent、Private exponent 和 Modulus：

```
def extended_gcd (a, b): // return d, x, y such that ax + by = d
    if b == 0:
        return (a, 1, 0);
    (d, x, y) = extended_gcd(b, a % b);
    return (d, y, x - (a / b) * y);

N = p * q
```

```
r = (p - 1) * (q - 1)
e = 65537
assert gcd(e, r) == 1
(a, d, b) = extended_gcd(e, r)
print("Public exponent: ", e)
print("Private exponent: ", d)
print("Modulus: ", N)
```

範例輸入 範例輸出

提示：為了方便實作，測資皆可以用 65537 當作 public exponent。當然你可以自行任意產生公鑰跟私鑰，只要能夠用於正常解密加密即可。

加密模式 (5+5%)

給定數字 n、Public exponent e 和 Modulus N，可以用以下公式計算 n 透過 e 和 N 加密後的結果。

$$c = n^e \pmod{N}$$

以下 pseudo code 會計算數字 n 透過 Public exponent e 和 Modulus N 加密後的結果。

```
def power(a, n, N):
    ans = 1;
    while n > 0:
        if n % 2 == 1:
            ans = ans * a % N;
        a = a * a % N;
        n = floor(n / 2)
    return ans;

print(power(n, e, N))
```

注意這份 pseudo code 不包含檔案的讀取跟處理，相關的程式碼請自行實作。另一種輸入方式的加密也請同學自行實作。testfile 範例輸入 1 範例輸入 2 範例輸出

提示：兩種輸入方式各佔五分

解密模式 (5%)

給定數字 c、Private exponent d 和 Modulus N，可以用以下公式計算 c 透過 d 和 N 解密後的結果。

$$n = c^d \pmod{N}$$

解密的原理是已知 $ed \equiv 1(\mod r)$，也就是 $ed = 1 + h\varphi(N)$，於是我們有

$$n^{ed} = n^{1+h\varphi(N)} = n \cdot n^{h\varphi(N)} = n \cdot (n^{\varphi(N)})^h$$

如果 n 和 N 互質，那麼 $n^{ed} \equiv n \cdot (n^{\varphi(N)})^h \equiv n \cdot (1)^h \equiv n \pmod{N}$

否則，不失一般性的令 $n = ph$，且 $ed - 1 = q(k-1)$，那麼

$$n^{ed} \equiv (ph)^{ed}0 \equiv n \pmod{p}$$

$$n^{ed} \equiv n^{ed-1} \cdot n \equiv n^{q(k-1)} \cdot n \equiv (n^{q-1})^k \cdot n \equiv 1^k \cdot n \equiv n \pmod{q}$$

綜合以上兩式，即可得到 $n^{ed} \equiv n \pmod{N}$。

以下 pseudo code 會計算數字 c 透過 Private exponent d 和 Modulus N 解密後的結果。

```
def power(a, n, N):
    ans = 1;
    while n > 0:
        if n % 2 == 1:
            ans = ans * a % N;
        a = a * a % N;
        n = floor(n / 2)
    return ans;

print(power(c, d, N))
```

注意這份 pseudo code 不包含檔案的讀取跟處理，相關的程式碼請自行實作。

testfile 範例輸入 範例輸出

II. Do I know this DSA? (5%)

在你實作了 RSA 的演算法後，NASA 的系統變得極度安全，那些酷酷的外國人再也沒有辦法攻擊任何服務了，但是由於太過安全了，連其他的 NASA 成員都沒有辦法進到服務裡面了，因此希望你再寫一個數位簽章的演算法 (DSA)，來協助驗證使用者的身份。

1. Verify a signature

在 DSA 的驗證環節中，你會獲得訊息 x，簽章對 (r, s) 以及驗證公鑰 (p, q, α, β)，並需要執行以下五個步驟對獲得的訊息 x 進行驗證：

1. 計算 $w \equiv s^{-1} \pmod{q}$
2. 計算 $u_1 \equiv w \times \text{SHA}(x) \pmod{q}$
3. 計算 $u_2 \equiv w \times r \pmod{q}$
4. 計算 $v \equiv (\alpha^{u_1} \times \beta^{u_2} \mod p) \pmod{q}$
5. 判斷 v 是否等於 r，若是則代表驗證成功

對於驗證成功者，輸出 “True”；驗證失敗者輸出 “False”。
提示：本題使用的 SHA 函數可以用 Linux 內建的 sha1sum 指令計算完成

```
This command is an DSA verifying implementation written in shell script.
usage: ./p2.sh --sig (<r>,<s>) --pubkey (<p>,<q>,<alpha>,<beta>) <file>
```

指令內容

1. `-h --help` 輸出使用說明
2. `--sig` 會接著輸入簽章對 (r, s)

3. `--pubkey` 會接著輸入公鑰對 (p, q, α, β)
4. `<file>` 代表需要驗證的檔案

補充事項

1. 保證本題輸入數字皆合法
2. 給定的數字輸入可能為十六進位表示：以 "0x" 開頭

testfile 範例輸入 範例輸出