

Problem Description

當你點開這個題目後，你應該發現了一件事情...
這個 Judge 沒有上傳按鈕！

使用 Git 是資工人該學會的技能，因此這次我們決定使用 Git 來作為 Judge 的上傳管道！

當你學會使用這個 Judge 的上傳方式後，你就會自動獲得這題的 AC Code ！

Git 安裝

各種作業系統上的 Git 安裝方式，請參閱以下連結

<https://git-scm.com/book/zh-tw/v2/%E9%96%8B%E5%A7%8B-Git-%E5%AE%89%E8%A3%9D%E6%95%99%E5%AD%B8>

安裝完成之後，建議使用 Windows 的同學使用 Git Bash 來做接下來步驟的操作介面。

如何產生一把自己的 SSH Key

為了安全性，這個 Judge 的 Git 上傳方式僅提供使用 SSH Public Key 登入的方式。

如果想要知道 SSH Public Key 登入的詳細原理或優點，可以自行上網查詢各式資料，這裡不詳細描述。

首先你會先需要產生一把自己的 SSH Key，如果你已經有了，那請忽略這個步驟。

可以使用以下的指令，檢查自己是否已經有 SSH Key。

```
ls ~/.ssh
```

如果有看到成對的檔案(OOXX和OOXX.pub)，那就代表你已經有了，請忽略產生 SSH Key 的這個步驟。

以下的指令可以用來產生一把 SSH Key。

```
ssh-keygen
```

接下來他會詢問你一些問題，如果不清楚的話，可以直接使用預設值(按Enter)。

接下來使用以下的指令，把你 SSH Key 列印到螢幕上。

```
cat ~/.ssh/id_rsa.pub
```

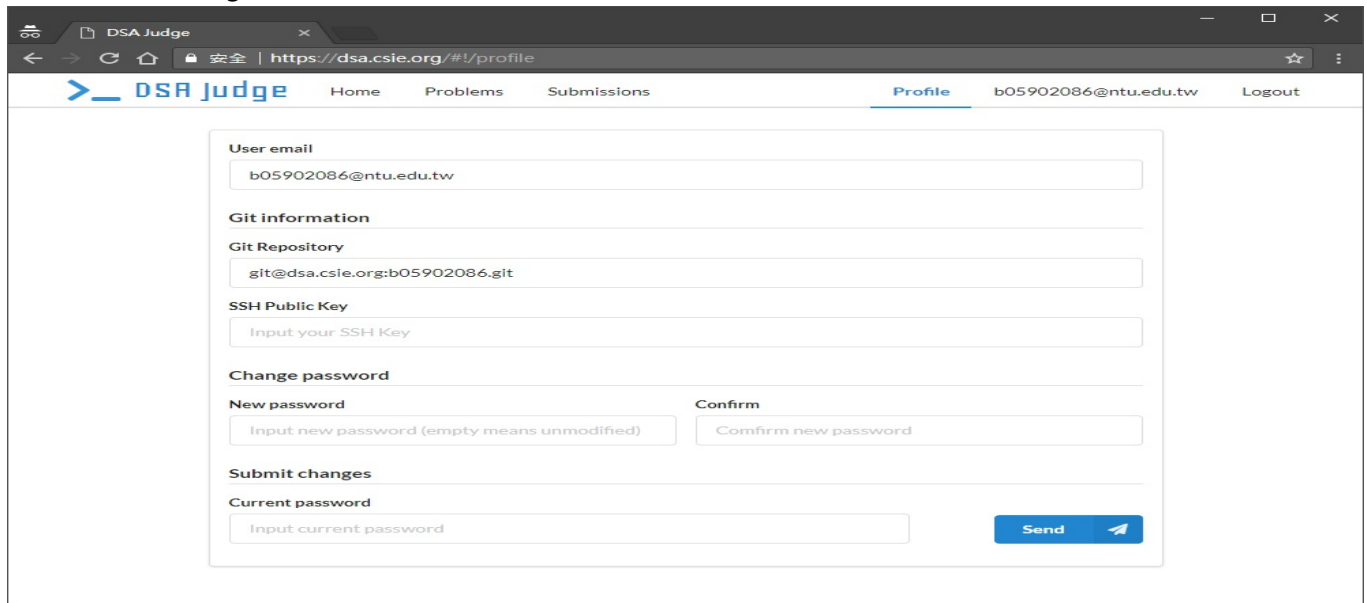
你會看到一串長的像類似底下的東西，這一大串文字就是你的 SSH Key。

```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAKl0UpkDHrfHY17SbrmTIpNLTKG9Tjom/BWDSU  
GP1+nafz1HDTYW7hdI4yZ5ew18JH4JW9jbhUFRviQzM7x1ELEVF4h91FX5QVkbPppSwg0cda3
```

```
Pbv7k0dJ/MTyBlWXFCR+HAo3FXRitBqxiX1nKhXpHAZsMciLq8V6RjsNAQwdsdMFvSlVK/7XA  
t3FaoJoAsncM1Q9x5+3V0Ww68/eIFmb1zuUF1jQJKprX88XypNDvjYNby6vw/Pb0rwert/En  
mZ+AW4OZPnTPI89ZPmVMLuayrD2cE86Z/il8b+gw3r3+1nKatmIkjn2so1d01QraTlMqVSsbx  
NrRFi9wrf+M7Q== schacon@agadorlaptop.local
```

如何使用 Git 來把程式傳到 Judge 上

接下來會看到 Judge 介面的右上方有 **Profile** 按鈕，點擊後進入個人的設定介面



The screenshot shows the DSA Judge Profile page. The page has a navigation bar with links: Home, Problems, Submissions, Profile (selected), and Logout. The Profile page contains several form sections: 'User email' with the value 'b05902086@ntu.edu.tw'; 'Git information' with 'Git Repository' set to 'git@dsa.csie.org:b05902086.git' and 'SSH Public Key' as an empty field; 'Change password' with 'New password' and 'Confirm' fields; and 'Submit changes' with a 'Current password' field. A blue 'Send' button is located at the bottom right of the form.

請把自己的 SSH Key 填入畫面中的 **SSH Public Key** 欄位，然後最下方的 **Current password** 請輸入你現在登入 Judge 使用的密碼，然後按下 **Send** 送出。

送出後畫面右下角應該會短暫跳出一個小框，如果是綠色的，代表成功了，如果是紅色的則代表有東西出錯，可以仔細看一下上面的訊息，並嘗試解決。

接下來注意到畫面上的 **Git Repository**，那是你要把寫好的程式送過去的地方。以下使用 **<Your Git Reop>** 代表這串文字。

接下來到你的電腦上，找一個你喜歡放程式的位置，打入以下的指令

```
git clone <Your Git Reop>
```

如果沒有意外的話，你應該已經看到一個以你的學號命名的資料夾。

打開後會發現裡面有一個名為 **0** 的資料夾，那資料夾中有個名為 **main.c** 的程式，你需要把他打開來做點修改(例如加點空白或換行之類的都行)，然後存檔。

接下來在名字為你的學號這個目錄中，輸入以下的git指令，把修改過的程式傳到 Judge，檢查是否正確。(其中 **some message**的內容可以自行修改)

```
git add .  
git commit -m "<some message>"  
git push
```

你在打 **git commit** 的時候有可能會噴出沒有設定 **git** 的使用者的訊息，請照著畫面上的指示操作。

接下來你應該能在 **Judge** 上看到上傳的結果

而 **git push** 中也會告訴你你上傳的 **Submission ID**，可以參考下面的圖片。

```
(ubuntu)joe@VAIO: ~/b05902086
(ubuntu)joe@VAIO:~[0]$ git clone git@dsa.csie.org:b05902086.git
Cloning into 'b05902086'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (4/4), done.
Checking connectivity... done.
(ubuntu)joe@VAIO:~[0]$ cd b05902086/
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ ls
.
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ cd 0
(ubuntu)joe@VAIO:~/b05902086/0 (master)[0]$ ls
main.c
(ubuntu)joe@VAIO:~/b05902086/0 (master)[0]$ vim main.c
(ubuntu)joe@VAIO:~/b05902086/0 (master)[0]$ cd ..
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git add .
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git commit -m "add some space"
[master fb7e52a] add some space
1 file changed, 1 insertion(+), 1 deletion(-)
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git push
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 368 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote: You submit Problem #0 !
remote: Problem #0 not found.
To git@dsa.csie.org:b05902086.git
355f576..fb7e52a master -> master
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$
```

```
(ubuntu)joe@VAIO: ~/b05902086
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ mkdir 39
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ vim 39/main.c
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git add .
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git commit -m "write 39"
[master 5810dad] write 39
1 file changed, 1 insertion(+)
create mode 100644 39/main.c
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git push
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 354 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote: You submit Problem #39 !
remote: Submission: https://dsa.csie.org/#!/submission/186
To git@dsa.csie.org:b05902086.git
79b336e..5810dad master -> master
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$
```

```
(ubuntu)joe@VAIO: ~/b05902086
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ vim 40/main.c
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git add .
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git commit -m update
[master b09f944] update
1 file changed, 1 insertion(+), 1 deletion(-)
(ubuntu)joe@VAIO:~/b05902086 (master)[0]$ git push
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 387 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote: You submit Problem #40 !
remote: Problem #40 quota used up.
To git@dsa.csie.org:b05902086.git
6c1ad44..b09f944 master -> master
```

以上圖片為開發中畫面，內容僅供參考。

注意事項

1. 如果你沒有在 Judge 上面的 Submission 中看到，就是代表沒有繳交紀錄。
2. 每人每天每題最多只能上傳五次，無法累積。
3. Git 上傳方式只會檢查你對於 master 分支上的修改，其他分支上的修改都不會上傳至 Judge。
4. Git 上傳方式只會上傳你對於 master 分支上有符合正確位置的程式碼(<problem_id>/main.c)，其於的東西皆不會傳到Judge上。
5. Git 上傳方式只會上傳你上次 push 跟這次 push 間對於 master 分支上有做修改的檔案，未做修改的程式碼不會上傳。
6. 對於已經傳上 master 分支的 commit，請勿對其做 rebase 之類會影響commit的樹狀結構的操作，否則你很可能會傳不上來。
7. 請不要把太多的大型垃圾 commit 上來，否則如過超過了使用上限，你可能之後的東西會 push 不上來。