

Name: Thái Vĩnh Đạt

ID: 22520235

Class: IT007.O21.ANTN

OPERATING SYSTEM LAB X'S REPORT

SUMMARY

Task		Status	Page
Section 2.5	1	Done	2
	2	Done	9
	3	Done	9
	4	done	10
Section 2.6	1	Done	11
	2	Done	12

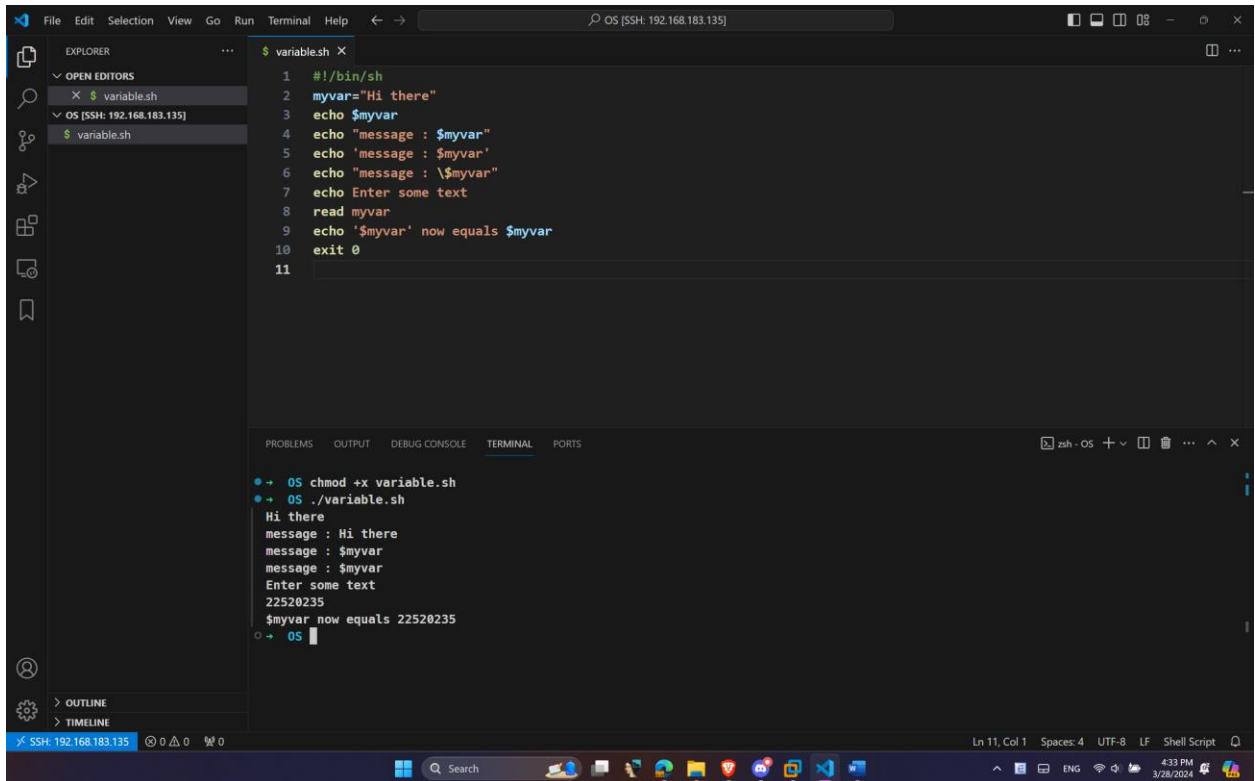
Self-scores:

Note: Export file to **PDF and name the file by following format:
LAB X – <Student ID>.pdf*

Section 2.5

1.

- Ví dụ 2-1:



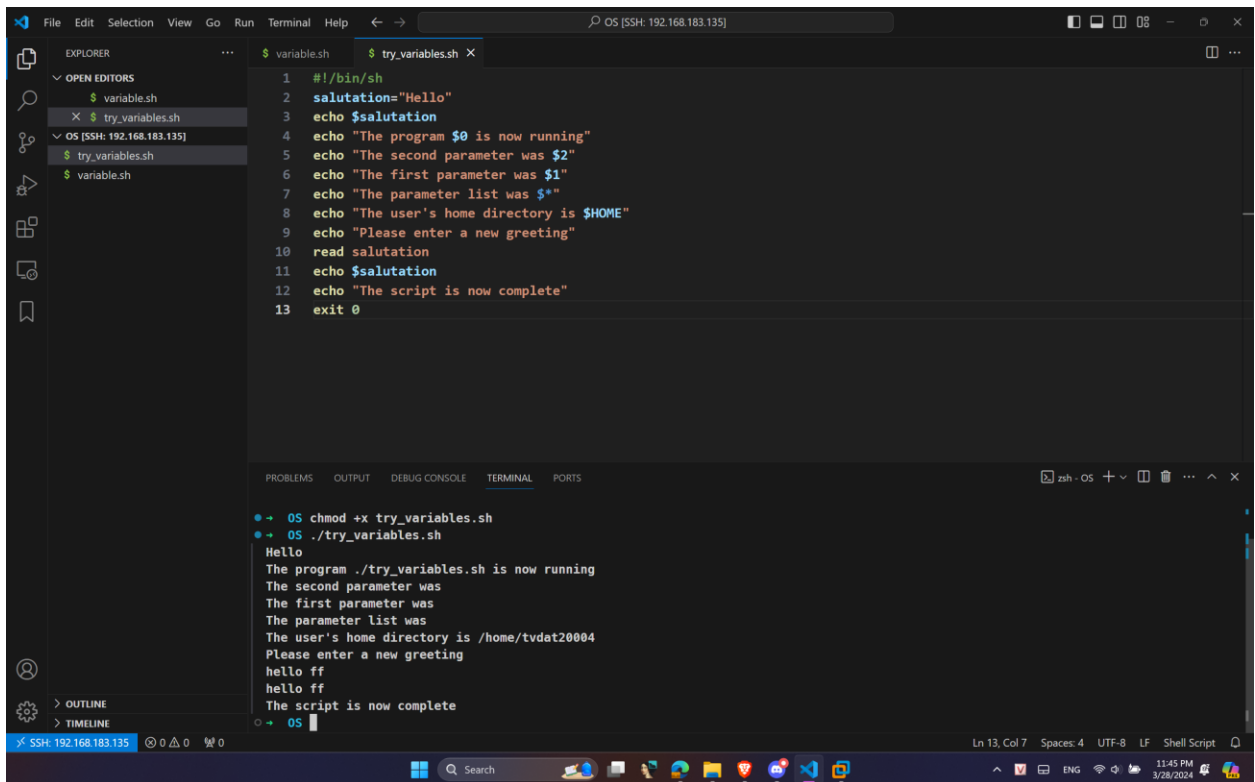
The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a file named `variable.sh` in the `OS [SSH: 192.168.183.135]` workspace. The editor displays the following shell script:

```
1 #!/bin/sh
2 myvar="Hi there"
3 echo $myvar
4 echo "message : $myvar"
5 echo 'message : $myvar'
6 echo "message : \myvar"
7 echo Enter some text
8 read myvar
9 echo '$myvar' now equals $myvar
10 exit 0
11
```

The terminal at the bottom shows the execution of the script:

```
OS chmod +x variable.sh
OS ./variable.sh
Hi there
message : Hi there
message : $myvar
message : $myvar
message : $myvar
Enter some text
22520235
$myvar now equals 22520235
OS
```

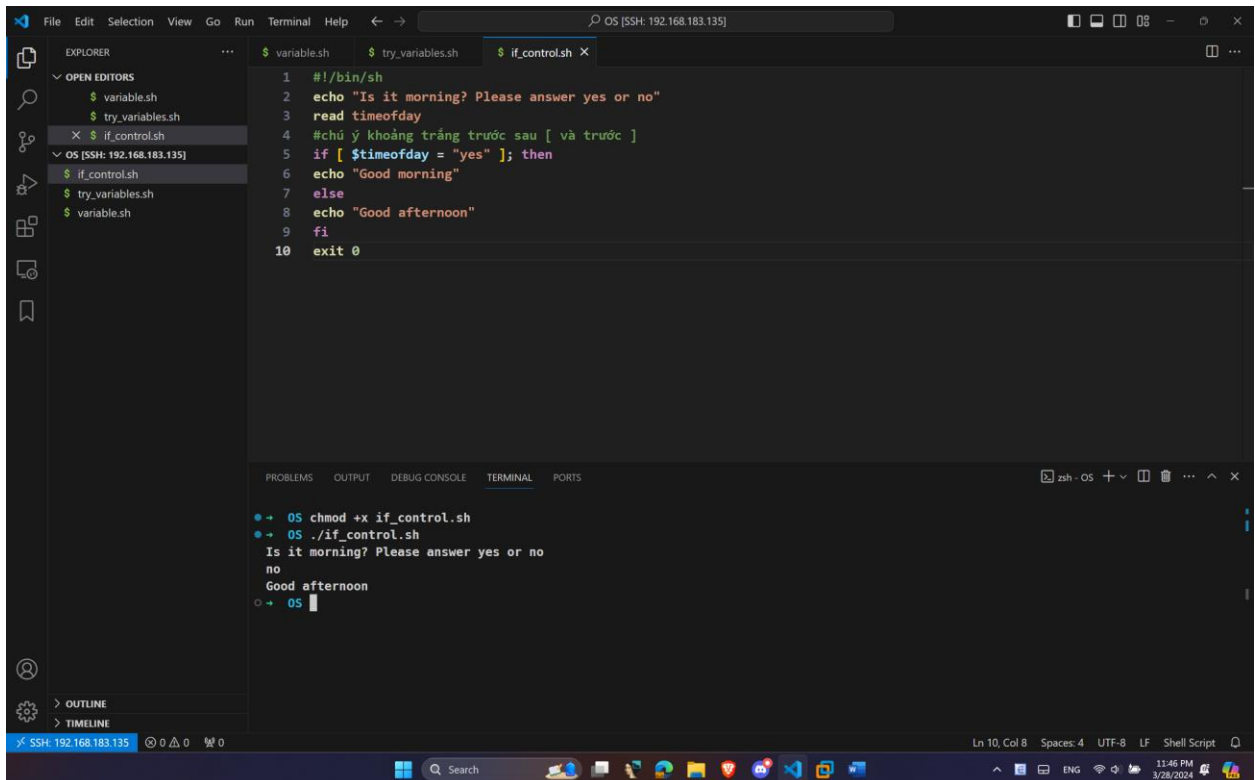
- Ví dụ 2-2:



```
1 #!/bin/sh
2 salutation="Hello"
3 echo $salutation
4 echo "The program $0 is now running"
5 echo "The second parameter was $2"
6 echo "The first parameter was $1"
7 echo "The parameter list was $"
8 echo "The user's home directory is $HOME"
9 echo "Please enter a new greeting"
10 read salutation
11 echo $salutation
12 echo "The script is now complete"
13 exit 0
```

```
OS chmod +x try_variables.sh
OS ./try_variables.sh
Hello
The program ./try_variables.sh is now running
The second parameter was
The first parameter was
The parameter list was
The user's home directory is /home/tvdat20004
Please enter a new greeting
hello ff
hello ff
The script is now complete
OS
```

- Ví dụ 2-3:



```
1 #!/bin/sh
2 echo "Is it morning? Please answer yes or no"
3 read timeofday
4 #chữ ý khoảng trắng trước sau [ và trước ]
5 if [ $timeofday = "yes" ]; then
6 echo "Good morning"
7 else
8 echo "Good afternoon"
9 fi
10 exit 0
```

```
OS chmod +x if_control.sh
OS ./if_control.sh
Is it morning? Please answer yes or no
no
Good afternoon
OS
```

- Ví dụ 2-4:

```
1 #!/bin/sh
2 echo "Is it morning? Please answer yes or no"
3 read timeofday
4 if [ $timeofday = "yes" ]; then
5     echo "Good morning"
6 elif [ $timeofday = "no" ]; then
7     echo "Good afternoon"
8 else
9     echo "Sorry, $timeofday not recognized. Enter
10 yes or no"
11 exit 1
12 fi
13 exit 0
```

```
OS chmod +x elif_control.sh
OS ./elif_control.sh
Is it morning? Please answer yes or no
no
Good afternoon
OS ./elif_control.sh
Is it morning? Please answer yes or no
hehe
Sorry, hehe not recognized. Enter
yes or no
OS
```

- Ví dụ 2-5

```
1 #!/bin/sh
2 echo -n "Is it morning? Please answer yes or no: "
3 read timeofday
4 if [ "$timeofday" = "yes" ]; then
5     echo "Good morning"
6 elif [ "$timeofday" = "no" ]; then
7     echo "Good afternoon"
8 else
9     echo "Sorry, $timeofday not recognized. Enter yes or
10 no"
11 exit 1
12 fi
13 exit 0
```

```
OS ./elif_control.sh
Is it morning? Please answer yes or no
hehe
Sorry, hehe not recognized. Enter
yes or no
OS chmod +x elif_control2.sh
OS ./elif_control2.sh
Is it morning? Please answer yes or no:
Sorry, not recognized. Enter yes or
no
OS
```

- Ví dụ 2-6

The screenshot shows the Visual Studio Code interface with a file explorer on the left and a terminal at the bottom. The file explorer shows a list of files: `variable.sh`, `try_variables.sh`, `if_control.sh`, `elif_control.sh`, `elif_control2.sh`, and `for_loop.sh`. The `for_loop.sh` file is open in the editor, showing the following script:

```
1 #!/bin/sh
2 for foo in bar fud 13
3 do
4 echo $foo
5 done
6 exit 0
```

The terminal at the bottom shows the execution of the script:

```
OS chmod +x for_loop.sh
OS ./for_loop.sh
bar
fud
13
OS
```

- Ví dụ 2-7

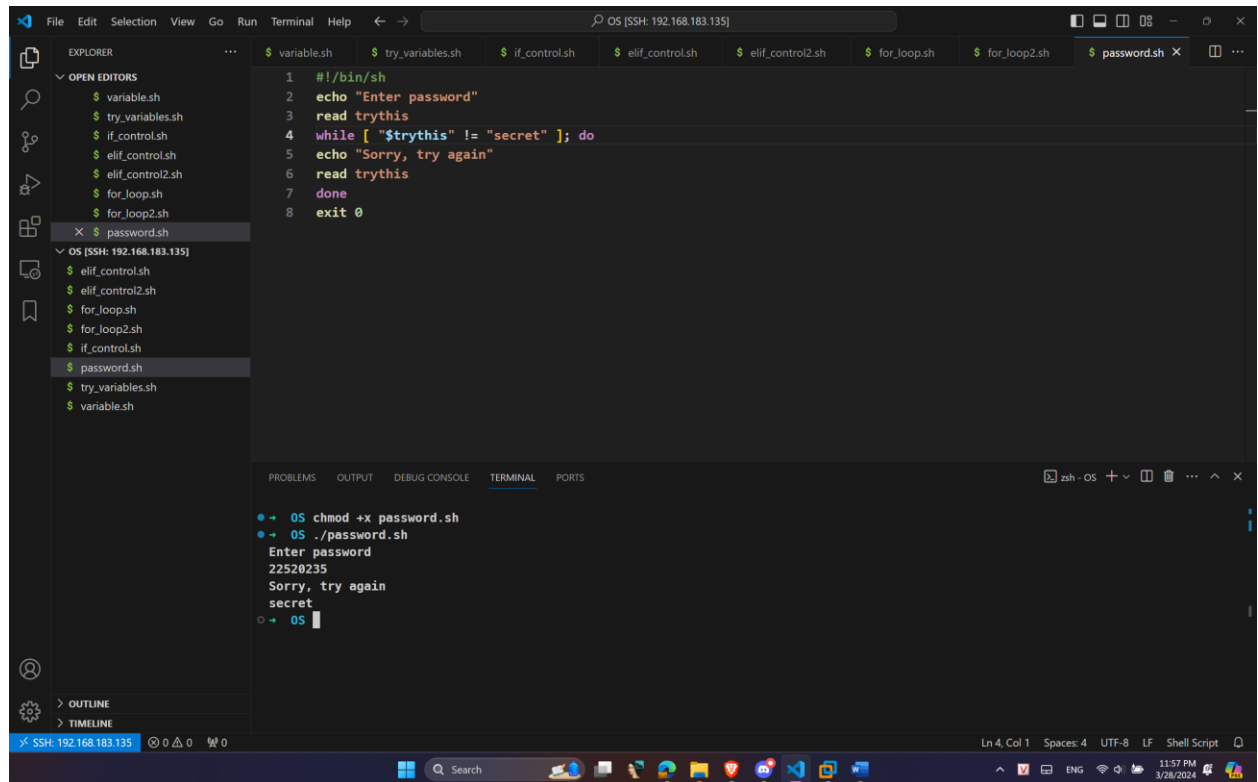
The screenshot shows the Visual Studio Code interface with a file explorer on the left and a terminal at the bottom. The file explorer shows a list of files: `variable.sh`, `try_variables.sh`, `if_control.sh`, `elif_control.sh`, `elif_control2.sh`, `for_loop.sh`, and `for_loop2.sh`. The `for_loop2.sh` file is open in the editor, showing the following script:

```
1 #!/bin/sh
2 for file in $(ls f*.sh); do
3 more $file
4 done
```

The terminal at the bottom shows the execution of the script:

```
OS chmod +x for_loop2.sh
OS ./for_loop2.sh
#!/bin/sh
for file in $(ls f*.sh); do
more $file
done
#!/bin/sh
for foo in bar fud 13
do
echo $foo
done
exit 0
OS
```

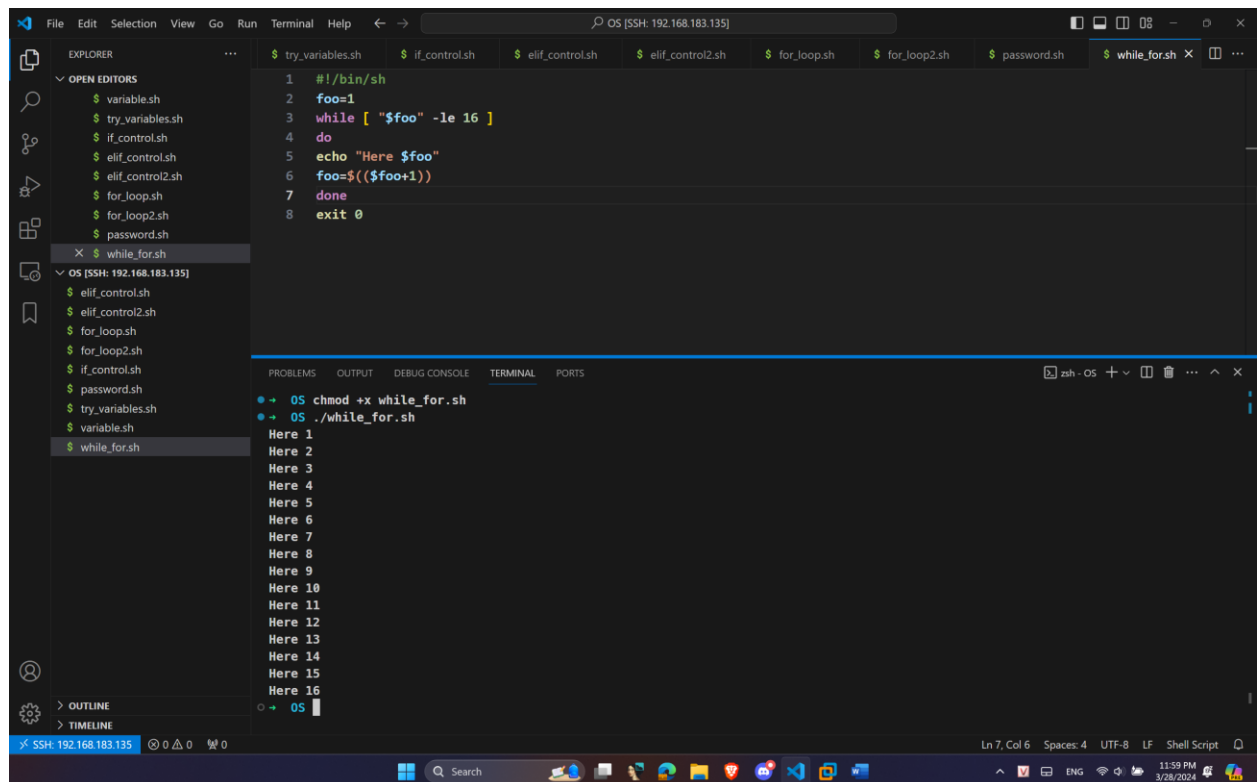
- Ví dụ 2-8



```
1 #!/bin/sh
2 echo "Enter password"
3 read trythis
4 while [ "$trythis" != "secret" ]; do
5 echo "Sorry, try again"
6 read trythis
7 done
8 exit 0
```

```
OS [SSH: 192.168.183.135]
OS chmod +x password.sh
OS ./password.sh
Enter password
22520235
Sorry, try again
secret
OS
```

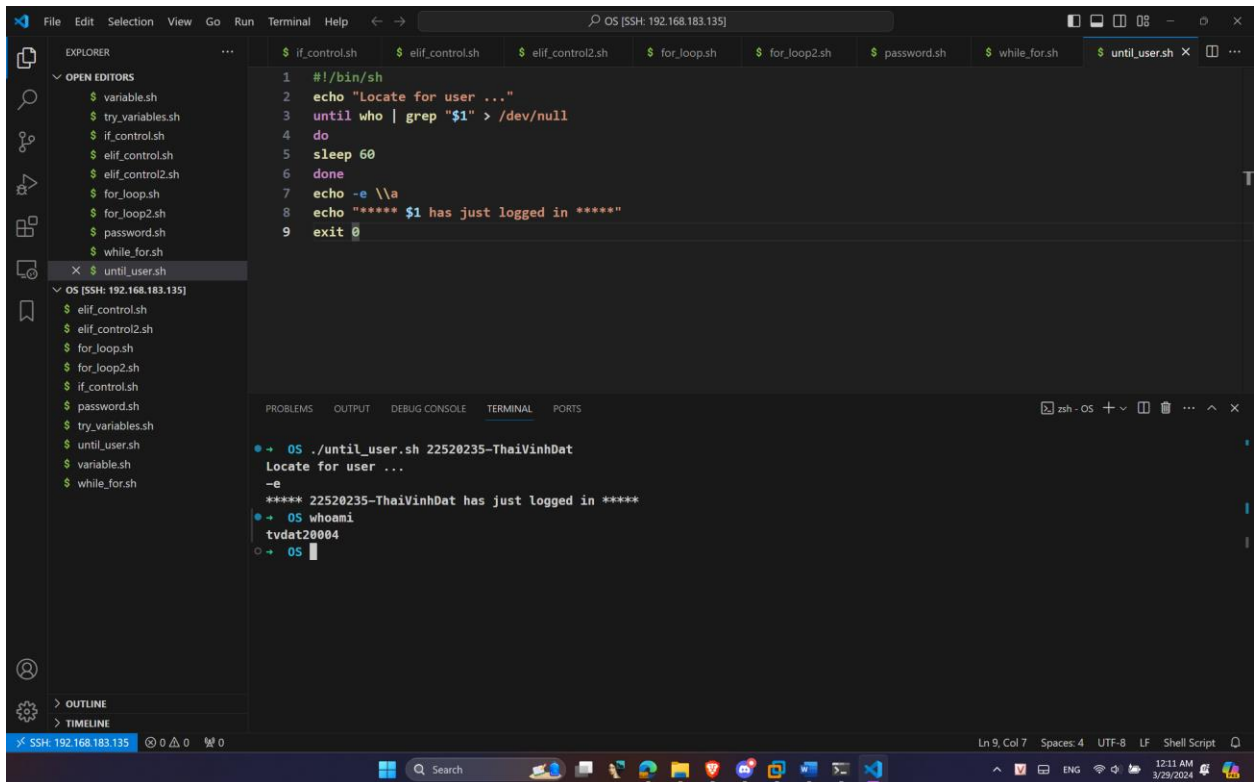
- Ví dụ 2-9



```
1 #!/bin/sh
2 foo=1
3 while [ "$foo" -le 16 ]
4 do
5 echo "Here $foo"
6 foo=$((foo+1))
7 done
8 exit 0
```

```
OS [SSH: 192.168.183.135]
OS chmod +x while_for.sh
OS ./while_for.sh
Here 1
Here 2
Here 3
Here 4
Here 5
Here 6
Here 7
Here 8
Here 9
Here 10
Here 11
Here 12
Here 13
Here 14
Here 15
Here 16
OS
```

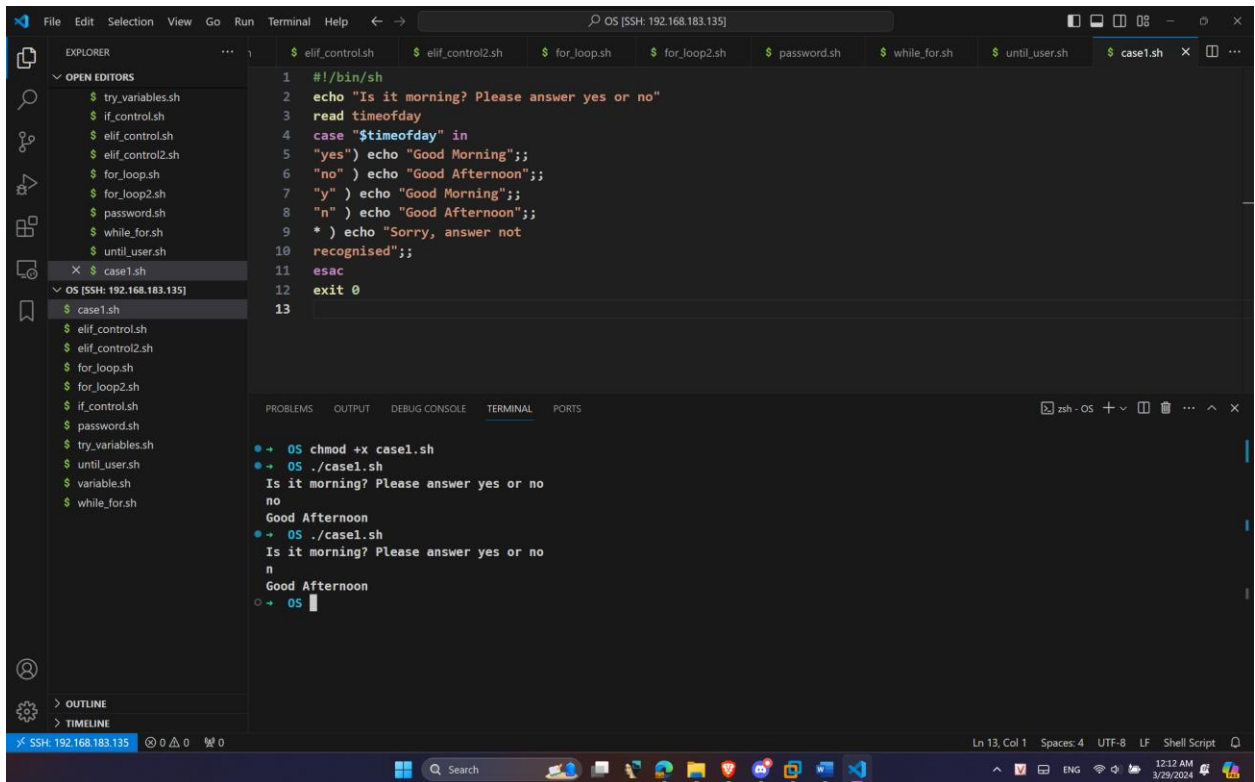
- Ví dụ 2-10



```
1 #!/bin/sh
2 echo "Locate for user ..."
3 until who | grep "$1" > /dev/null
4 do
5     sleep 60
6 done
7 echo -e \\a
8 echo "***** $1 has just logged in *****"
9 exit 0
```

```
OS [SSH: 192.168.183.135]
$ ./until_user.sh 22520235-ThaiVinhDat
Locate for user ...
***** 22520235-ThaiVinhDat has just logged in *****
$ whoami
tvdatt20004
$
```

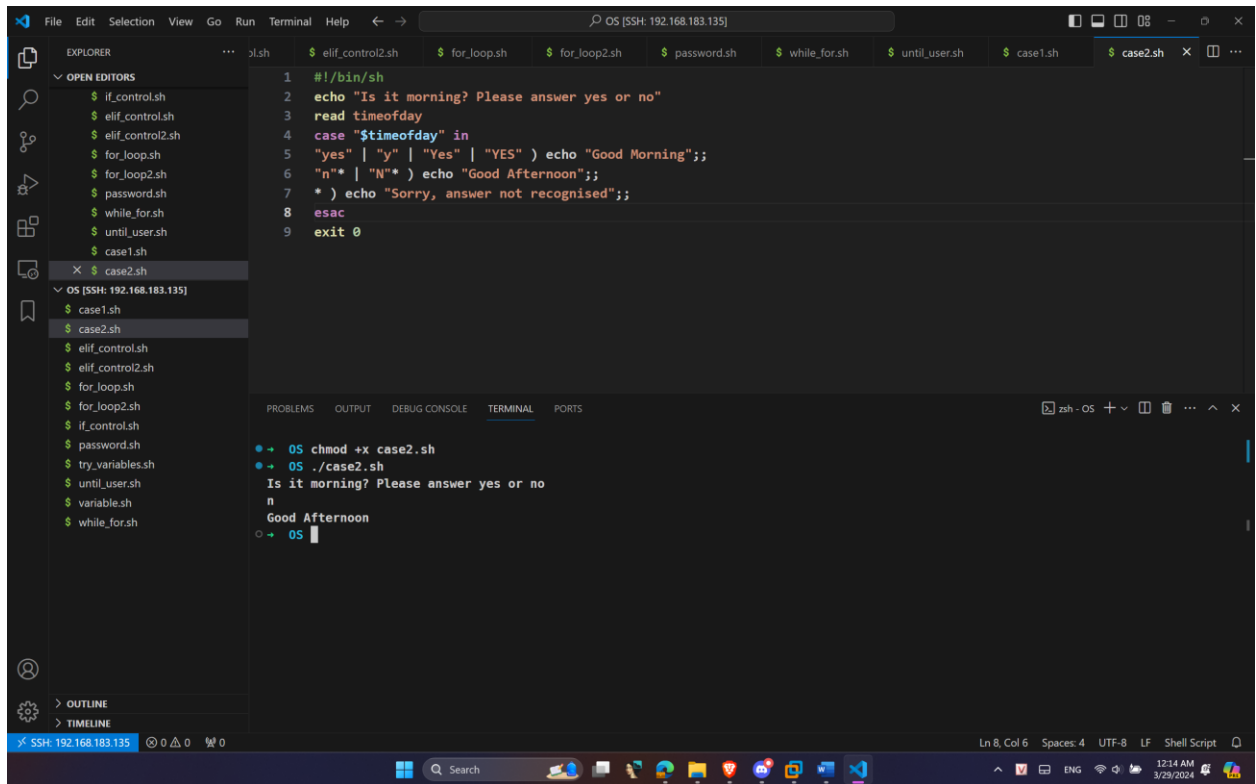
- Ví dụ 2-11



```
1 #!/bin/sh
2 echo "Is it morning? Please answer yes or no"
3 read timeofday
4 case "$timeofday" in
5     "yes") echo "Good Morning";;
6     "no" ) echo "Good Afternoon";;
7     "y" ) echo "Good Morning";;
8     "n" ) echo "Good Afternoon";;
9     * ) echo "Sorry, answer not
10 recognised";;
11 esac
12 exit 0
13
```

```
OS [SSH: 192.168.183.135]
$ case1.sh
Is it morning? Please answer yes or no
no
Good Afternoon
$ ./case1.sh
Is it morning? Please answer yes or no
n
Good Afternoon
$
```

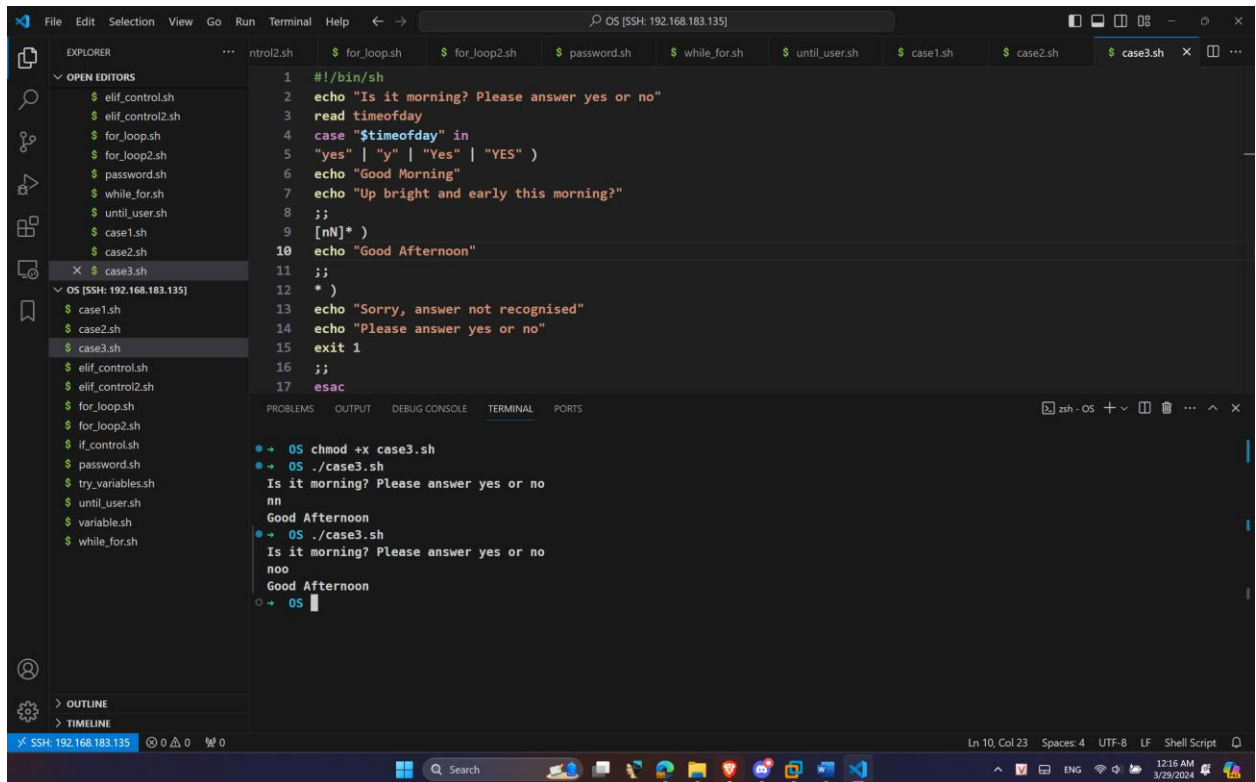
- Ví dụ 2-12



```
1 #!/bin/sh
2 echo "Is it morning? Please answer yes or no"
3 read timeofday
4 case "$timeofday" in
5 "yes" | "y" | "Yes" | "YES" ) echo "Good Morning";;
6 "n"* | "N"* ) echo "Good Afternoon";;
7 * ) echo "Sorry, answer not recognised";;
8 esac
9 exit 0
```

```
OS chmod +x case2.sh
OS ./case2.sh
Is it morning? Please answer yes or no
n
Good Afternoon
OS
```

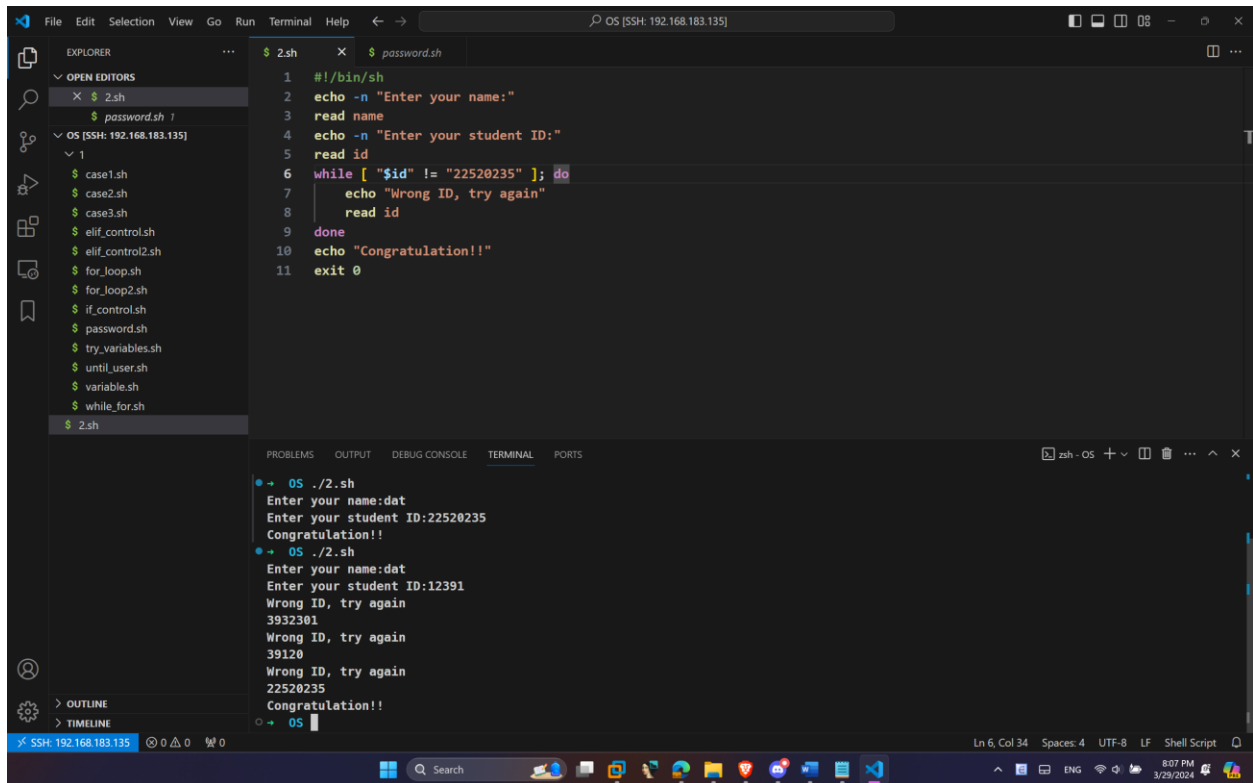
- Ví dụ 2-13:



```
1 #!/bin/sh
2 echo "Is it morning? Please answer yes or no"
3 read timeofday
4 case "$timeofday" in
5 "yes" | "y" | "Yes" | "YES" )
6 echo "Good Morning"
7 echo "Up bright and early this morning?"
8 ;;
9 [nN]* )
10 echo "Good Afternoon"
11 ;;
12 * )
13 echo "Sorry, answer not recognised"
14 echo "Please answer yes or no"
15 exit 1
16 ;;
17 esac
```

```
OS chmod +x case3.sh
OS ./case3.sh
Is it morning? Please answer yes or no
nn
Good Afternoon
OS ./case3.sh
Is it morning? Please answer yes or no
noo
Good Afternoon
OS
```


2.



The screenshot shows a Visual Studio Code editor window with a terminal session. The Explorer panel on the left lists several shell scripts, including `password.sh`. The main editor area displays the content of `password.sh`, which is a script that prompts for a name and a student ID, then checks if the ID is "22520235". If the ID is correct, it prints "Congratulation!!"; otherwise, it prints "Wrong ID, try again" and loops back. The terminal at the bottom shows the execution of the script twice. The first run uses the correct ID "22520235" and results in a congratulatory message. The second run uses the incorrect ID "12391" and results in the "Wrong ID, try again" message being printed multiple times, demonstrating the loop.

```
1 #!/bin/sh
2 echo -n "Enter your name:"
3 read name
4 echo -n "Enter your student ID:"
5 read id
6 while [ "$id" != "22520235" ]; do
7     echo "Wrong ID, try again"
8     read id
9 done
10 echo "Congratulation!!"
11 exit 0
```

```
OS ./2.sh
Enter your name:dat
Enter your student ID:22520235
Congratulation!!

OS ./2.sh
Enter your name:dat
Enter your student ID:12391
Wrong ID, try again
3932301
Wrong ID, try again
39120
Wrong ID, try again
22520235
Congratulation!!

OS
```

Bài này ta sẽ sử dụng cấu trúc vòng lặp while. Đầu tiên chương trình yêu cầu người dùng nhập tên và ID, sau đó lệnh while sẽ liên tục kiểm tra giá trị ID có bằng “22520235” hay không. Nếu người dùng nhập sai, chương trình sẽ yêu cầu nhập lại cho đến khi đúng với chuỗi yêu cầu.

3.

The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left displays a file tree for a remote SSH connection to 192.168.183.135. The file list includes several shell scripts, with '3.sh' selected. The main editor area shows the content of '3.sh', which is a shell script that prompts the user to enter a number n (n < 10), checks if the input is valid, and calculates the sum of numbers from 1 to n using the formula $sum = \frac{n(n+1)}{2}$. The script uses a while loop for input validation. The Terminal panel at the bottom shows the execution of the script, with the user entering '3', '4', '5', and '10'. The output shows that for n=3, the sum is 6; for n=4, the sum is 10; for n=5, the sum is 15; and for n=10, the sum is 55. The status bar at the bottom indicates the current line is 10, column 7, in UTF-8 encoding, LF line endings, and Shell Script mode.

```
1 #!/bin/sh
2 echo -n "Enter a number n (n < 10)":
3 read n
4 while [ $n -lt 10 ]; do
5     echo -n "Try again:"
6     read n
7 done
8 sum=$((n*(n+1)/2))
9 echo "Sum from 1 to n is $sum"
10 exit 0
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
OS ./3.sh
Enter a number n (n < 10):3
Try again:4
Try again:5
Try again:10
Sum from 1 to n is 55
OS
```

Ở đây, ta sẽ dùng phép lặp while để yêu cầu người dùng nhập số n, nếu $n < 10$ thì phải nhập lại cho đến khi lớn hơn hoặc bằng 10. Sau đó tính tổng từ 1 đến n bằng công thức $sum = \frac{n(n+1)}{2}$. Sau đó in ra biến sum.

4.

The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a file tree with a group named 'test.txt' containing several shell scripts. The main editor area has two tabs: 'test.txt' and '4.sh'. The 'test.txt' tab is active, showing a script that uses 'grep' to check if a string entered by the user is in 'test.txt'. The '4.sh' tab is also visible, showing the same script. The terminal at the bottom shows the execution of the script, with prompts for 'Enter a string:' and output messages indicating whether the string was found in 'test.txt'.

```
1 #!/bin/bash
2
3 echo -n "Enter a string:"
4 read string
5
6 if grep -q "$string" "test.txt"; then
7     echo "File test.txt contains '$string'"
8 else
9     echo "File test.txt does not contain '$string'"
10 fi
```

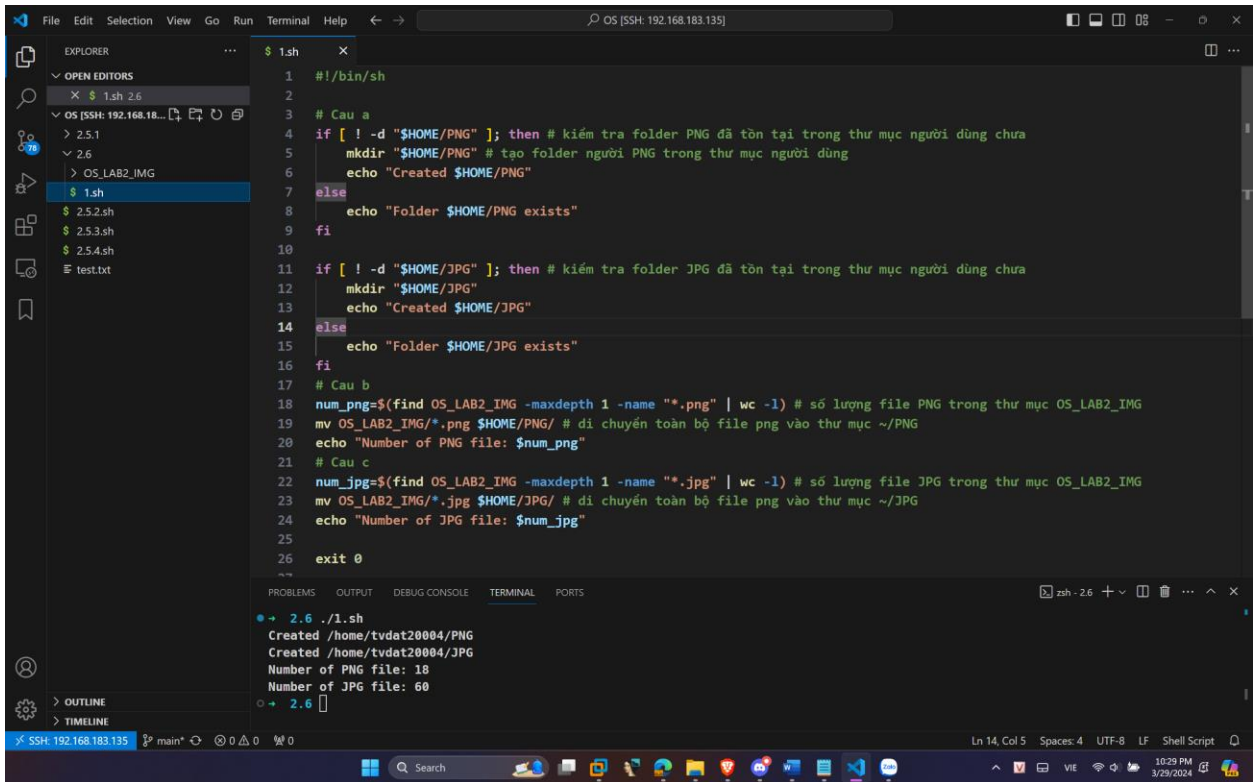
```
OS ./4.sh
Enter a string:well-chosen
File test.txt contains 'well-chosen'
OS ./4.sh
Enter a string:abcpz
File test.txt does not contain 'abcpz'
OS ./4.sh
Enter a string:12 year-old).
File test.txt contains '12 year-old).'
```

Bài này ta sẽ sử dụng điều kiện `grep -q "$string" "test.txt"` để kiểm tra xem biến string có trong file test.txt hay không, cụ thể:

- `grep`: Là một lệnh dùng để tìm kiếm một chuỗi hoặc một biểu thức chính quy trong một hoặc nhiều file.
- `-q`: Là một tùy chọn của lệnh `grep`, có nghĩa là “quiet” (im lặng). Khi sử dụng tùy chọn này, `grep` sẽ không in ra bất kỳ dòng nào trên màn hình. Thay vào đó, nó chỉ trả về giá trị thoát để cho biết liệu chuỗi đã được tìm thấy hay chưa.
- `$string`: là biến chứa chuỗi cần kiểm tra
- “test.txt”: là một file trong thư mục chứa code.

Section 2.6

1.



```
1 #!/bin/sh
2
3 # Cau a
4 if [ ! -d "$HOME/PNG" ]; then # kiểm tra folder PNG đã tồn tại trong thư mục người dùng chưa
5     mkdir "$HOME/PNG" # tạo folder người PNG trong thư mục người dùng
6     echo "Created $HOME/PNG"
7 else
8     echo "Folder $HOME/PNG exists"
9 fi
10
11 if [ ! -d "$HOME/JPG" ]; then # kiểm tra folder JPG đã tồn tại trong thư mục người dùng chưa
12     mkdir "$HOME/JPG" # tạo folder người JPG trong thư mục người dùng
13     echo "Created $HOME/JPG"
14 else
15     echo "Folder $HOME/JPG exists"
16 fi
17
18 # Cau b
19 num_png=$(find OS_LAB2_IMG -maxdepth 1 -name "*.png" | wc -l) # số lượng file PNG trong thư mục OS_LAB2_IMG
20 mv OS_LAB2_IMG/*.png $HOME/PNG/ # di chuyển toàn bộ file png vào thư mục ~/PNG
21 echo "Number of PNG file: $num_png"
22
23 # Cau c
24 num_jpg=$(find OS_LAB2_IMG -maxdepth 1 -name "*.jpg" | wc -l) # số lượng file JPG trong thư mục OS_LAB2_IMG
25 mv OS_LAB2_IMG/*.jpg $HOME/JPG/ # di chuyển toàn bộ file png vào thư mục ~/JPG
26 echo "Number of JPG file: $num_jpg"
27
28 exit 0
```

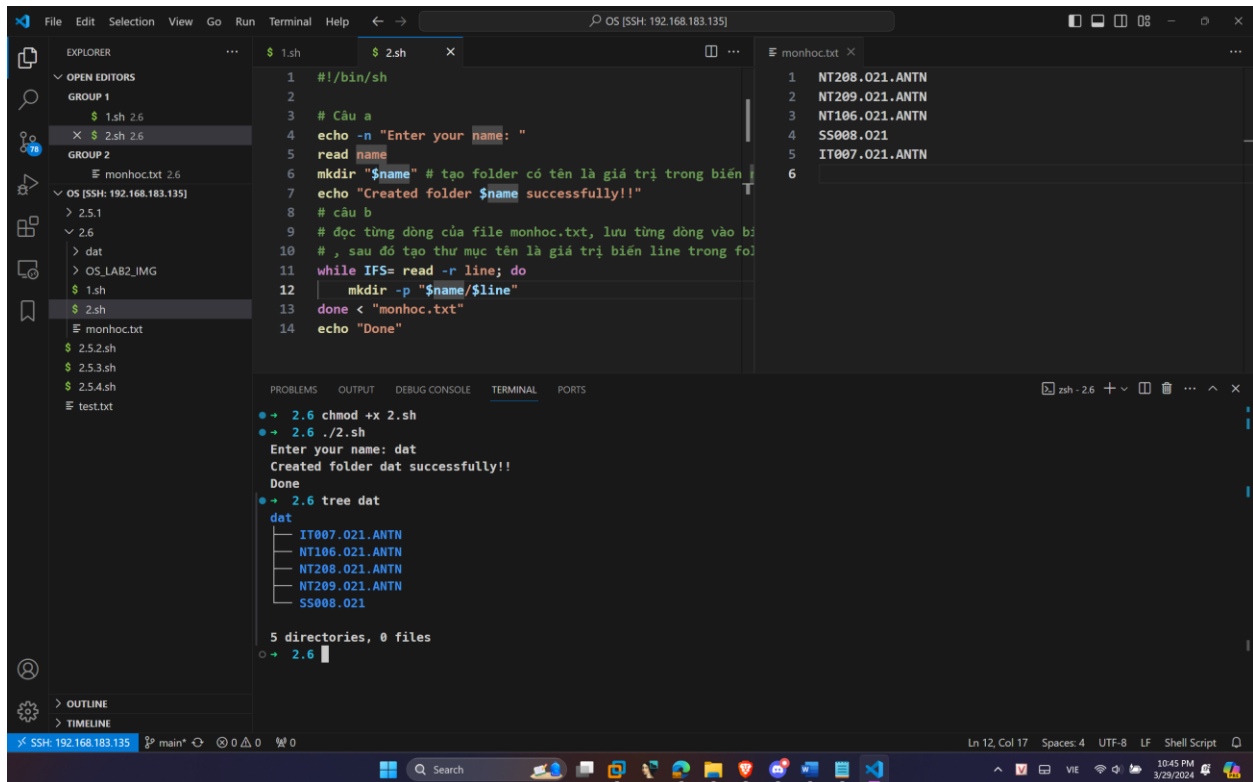
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - 2.6

Ln 14, Col 5 Spaces: 4 UTF-8 LF Shell Script

(Phân giải thích em đã chú thích trong code)

2.



```
1 #!/bin/sh
2
3 # Câu a
4 echo -n "Enter your name: "
5 read name
6 mkdir "$name" # tạo folder có tên là giá trị trong biến
7 echo "Created folder $name successfully!!"
8
9 # câu b
10 # , sau đó tạo thư mục tên là giá trị biến line trong file
11 while IFS= read -r line; do
12     mkdir -p "$name/$line"
13 done < "monhoc.txt"
14 echo "Done"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - 2.6

Ln 12, Col 17 Spaces: 4 UTF-8 LF Shell Script