

Họ và tên: Nguyễn Nguyên Ngọc Anh

Mã số sinh viên: 22520058

Lớp: IT007.O11.1 – VN(HT1)

HỆ ĐIỀU HÀNH BÁO CÁO LAB 1

CHECKLIST

2.5. BÀI TẬP THỰC HÀNH

	BT 1	BT 2	BT 3	BT 4
Trình bày cách làm		√	√	√
Chụp hình minh chứng	√	√	√	√
Giải thích kết quả		√	√	√

2.6. BÀI TẬP ÔN TẬP

	BT 1.a	BT 1.b	BT 1.c	BT 2.a	BT 2.b
Trình bày cách làm	√	√	√	√	√
Chụp hình minh chứng	√	√	√	√	√
Giải thích kết quả	√	√	√	√	√

Tư chấm điểm: 10

**Lưu ý: Xuất báo cáo theo định dạng PDF, đặt tên theo cú pháp:*

<MSSV>_LAB2.pdf

2.5. BÀI TẬP THỰC HÀNH

1. Chạy tất cả các đoạn lệnh ví dụ ở phần 2.4. Chụp hình kết quả chạy các file script và lưu vào báo cáo.

Ví dụ 2.1

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash variables.sh
Hi there
message : Hi there
message : \$myvar
messgae : \$myvar
Enter some text
anh
\$myvar now equals anh
- **nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile\$** █

Ví dụ 2.2

```
SHfile > $ try_variables.sh
```

```
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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash 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/nguyennguyenngocanh-22520058
Please enter a new greeting
anh
anh
The script is now complete
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$**

Ví dụ 2.3

```
SHfile > $ if_control.sh
1  #!/bin/sh
2  echo "Is it morning? Please answer yes or no"
3  read timeofday
4  #chú 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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash if_control.sh
Is it morning? Please answer yes or no
yes
Good morning
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash if_control.sh
Is it morning? Please answer yes or no
no
Good afternoon
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** █

Ví dụ 2.4

```
SHfile > $ elif_control.sh
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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash elif_control.sh
Is it morning? Please answer yes or no
ýe
Sorry, ýe not recognized. Enter
yes or no
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash elif_control.sh
Is it morning? Please answer yes or no
yes
Good morning
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash elif_control.sh
Is it morning? Please answer yes or no
no
Good afternoon

Ví dụ 2.5

```
SHfile > $ elif_control2.sh
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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash elif_control2.sh
Is it morning? Please answer yes or no: yes
Good morning
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash elif_control2.sh
Is it morning? Please answer yes or no: no
Good afternoon
- ⊗ **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash elif_control2.sh
Is it morning? Please answer yes or no: md
Sorry, md not recognized. Enter yes or
no
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$**

Ví dụ 2.6

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash for_loop.sh
bar
fud
13
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$**

Ví dụ 2.7

```
SHfile > $ for_loop2.sh
1  #!/bin/sh
2  for file in $(ls f*.sh); do
3  echo $file
4  done
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash for_loop2.sh
first.sh
for_loop.sh
for_loop2.sh
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** █

Ví dụ 2.8

```
SHfile > $ password.sh
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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● nguyennguyennngocanh-22520058@AndreNguyen:~/SHfile$ bash password.sh
Enter password
ngocanh
Sorry, try again
secret
○ nguyennguyennngocanh-22520058@AndreNguyen:~/SHfile$
```

Ví dụ 2.9

```
SHfile > $ while_for.sh
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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash 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
○ nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$
```

Ví dụ 2.10

```
SHfile > $ until_user.sh
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
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash until_user.sh xyz
Locate for user ...

***** xyz has just logged in *****

○ **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** □

```
SHfile > $ until_user.sh
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
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

○ **nguyennguyenngocanh-22520058@AndreNguyen:~\$** ssh xyz@172.22.116.212
xyz@172.22.116.212's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/advantage>

System information as of Mon Oct 16 22:41:05 +07 2023

System load:	0.04	Processes:	57
Usage of /:	0.3% of 1006.85GB	Users logged in:	1
Memory usage:	17%	IPv4 address for eth0:	172.22.116.212
Swap usage:	0%		

Ví dụ 2.11

```
SHfile > $ case1.sh
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 recognised";;
10 esac
11 exit 0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash case1.sh
Is it morning? Please answer yes or no
y
Good Morning
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash case1.sh
Is it morning? Please answer yes or no

Sorry, answer not recognised
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** █

Ví dụ 2.12

```
SHfile > $ case2.sh
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
6  Morning";;
7  "n"* | "N"* ) echo "Good
8  Afternoon";;
9  * ) echo "Sorry, answer
10 not recognised";;
11 esac
12 exit 0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash case2.sh
Is it morning? Please answer yes or no

Sorry, answer
not recognised
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash case2.sh
Is it morning? Please answer yes or no
y
Good
Morning
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash case2.sh
Is it morning? Please answer yes or no
n
Good
Afternoon
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** █

Ví dụ 2.13

```
SHfile > $ case3.sh
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
18  exit 0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
⊗ nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash case3.sh
Is it morning? Please answer yes or no

Sorry, answer not recognised
Please answer yes or no
● nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash case3.sh
Is it morning? Please answer yes or no
y
Good Morning
Up bright and early this morning?
● nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash case3.sh
Is it morning? Please answer yes or no
n
Good Afternoon
● nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash case3.sh
Is it morning? Please answer yes or no
nônnno
Good Afternoon
```


2. Viết chương trình cho phép nhập vào tên và MSSV. Kiểm tra nếu MSSV đó không trùng với mình thì bắt nhập lại. In ra màn hình kết quả.

- Cách làm :

```
#!/bin/bash
```

```
my_id="22520058"
```

```
echo "Enter your name: "
```

```
read name
```

```
echo "Enter your id: "
```

```
read id
```

```
while [ "$id" != "$my_id" ]
```

```
do
```

```
    echo "try again"
```

```
    echo "Enter your id again: "
```

```
    read id
```

```
done
```

```
echo "Hello, $name"
```

Ta sẽ cho 1 biến tạm để giữ MSSV , sau đó cho người dùng nhập tên và MSSV dùng vòng lặp while để kiểm tra mssv vừa nhập với biến tạm đã có sẵn nếu người dùng nhập sai, bắt người dùng nhập lại mssv đến khi đúng thì sẽ chào.

```
SHfile > $ practice2.sh
1  #!/bin/bash
2
3  my_id="22520058"
4
5  echo "Enter your name: "
6  read name
7  echo "Enter your id: "
8  read id
9  while [ "$id" != "$my_id" ]
10 do
11     echo "try again"
12     echo "Enter your id again: "
13     read id
14 done
15
16 echo "Hello, $name"
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash practice2.sh
Enter your name:
Ngoc Anh
Enter your id:
22520095
try again
Enter your id again:
22520058
Hello, Ngoc Anh
○ nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$
```

Dòng mã	Giải thích
#!/bin/bash	Đây là dòng đầu tiên của script và xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
my_id="22520058"	Khai báo một biến my_id và gán giá trị là MSSV của bạn, trong trường hợp này là "22520058".

echo "Enter your name: "	In ra màn hình thông báo yêu cầu người dùng nhập tên của họ.
read name	Đọc tên của người dùng từ bàn phím và gán giá trị cho biến name.
echo "Enter your id: "	In ra màn hình thông báo yêu cầu người dùng nhập MSSV.
read id	Đọc MSSV từ bàn phím và gán giá trị cho biến id.
while ["\$id" != "\$my_id"]	Bắt đầu một vòng lặp while, kiểm tra xem giá trị của biến id (MSSV người dùng đã nhập) có khác với my_id (MSSV của bạn) hay không.
echo "try again"	In ra thông báo yêu cầu người dùng nhập lại MSSV vì giá trị đã nhập không trùng với my_id.
echo "Enter your id again: "	In ra thông báo yêu cầu người dùng nhập lại MSSV.
read id	Đọc lại giá trị MSSV từ người dùng và gán giá trị cho biến id.
done	Kết thúc vòng lặp while khi MSSV nhập vào trùng với my_id.
echo "Hello, \$name"	In ra màn hình thông báo chào hỏi người dùng bằng tên mà họ đã nhập sau khi MSSV đã nhập vào trùng với my_id.

3. Viết chương trình cho phép nhập vào một số n. Kiểm tra nếu $n < 10$ thì yêu cầu nhập lại. Tính tổng các số từ 1 đến n. In kết quả ra màn hình.

- **Cách làm:**

```
#!/bin/bash
```

```
echo "Enter n"
read n;

while [ $n -lt 10 ]; do
    echo "n must greater than 10,please try again"
    read n
done

sum=0

for ((i = 1; i <= n; i++))
do
    sum=$((sum + i))
done

echo "sum from 1 to $n is: "
echo $sum
```

Chương trình sẽ yêu cầu người dùng nhập một số nguyên n, sau đó kiểm tra xem n có lớn hơn hoặc bằng 10 không. Nếu n không đạt yêu cầu, chương trình yêu cầu người dùng nhập lại. Nếu n đủ lớn, chương trình tính tổng từ 1 đến n và in ra màn hình.

```
$ practice3.sh X
SHfile > $ practice3.sh
1  #!/bin/bash
2
3  echo "Enter n"
4  read n;
5
6  while [ $n -lt 10 ]; do
7      echo "n must greater than 10,please try again"
8      read n
9  done
10
11  sum=0
12
13  for ((i = 1; i <= n; i++))
14  do
15      sum=$((sum + i))
16  done
17
18  echo "sum from 1 to $n is: "
19  echo $sum

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
• nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash practice3.sh
Enter n
8
n must greater than 10,please try again
20
sum from 1 to 20 is:
210
• nguyennnguyenngocanh-22520058@AndreNguyen:~/SHfile$
```

Đoạn mã	Giải thích
#!/bin/bash	Dòng đầu tiên xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
echo "Enter n"	Sử dụng lệnh echo để in ra màn hình thông báo yêu cầu người dùng nhập giá trị cho biến n.
read n;	Sử dụng lệnh read để đọc giá trị mà người dùng nhập từ bàn phím và lưu vào biến n.

while [\$n -lt 10]; do	Bắt đầu một vòng lặp while để kiểm tra xem giá trị của n có bé hơn 10 không. Nếu n bé hơn 10, vòng lặp sẽ tiếp tục.
echo "n must greater than 10,please try again"	In ra màn hình thông báo lỗi cho người dùng nếu n không đạt yêu cầu.
read n	Người dùng được yêu cầu nhập lại giá trị cho n nếu giá trị trước đó không đạt yêu cầu.
done	Kết thúc vòng lặp while.
sum=0	Khởi tạo biến sum với giá trị ban đầu là 0.
for ((i = 1; i <= n; i++)); do	Bắt đầu vòng lặp for để tính tổng từ 1 đến n. Vòng lặp này sẽ chạy từ i = 1 đến i <= n, tăng i lên 1 sau mỗi lần lặp.
sum=\$((sum + i))	Cộng giá trị i vào biến sum để tính tổng.
done	Kết thúc vòng lặp for.
echo "sum from 1 to \$n is: "	In ra màn hình thông báo về việc in tổng từ 1 đến n.
echo \$sum	In ra màn hình giá trị tổng sum.

4. Viết trình cho phép nhập vào một chuỗi. Kiểm tra chuỗi đó có tồn tại trong một file text (ví dụ test.txt) cùng thư mục hay không?

- **Cách làm:**

```
#!/bin/bash
```

```
echo "Enter the string to check :"  
read string
```

```
file="test.txt"
```

```
if [ -f "$file" ] ; then  
    if grep -q "$string" "$file"; then
```

```
    echo "String '$string' exist in $file"
else
    echo "String '$string' doesn't exist in $file"
fi
else
    echo "$file doesn't exist"
fi
```

Kết quả của đoạn mã này là kiểm tra xem chuỗi string có tồn tại trong tệp file không, và sau đó in ra màn hình thông báo tùy thuộc vào kết quả kiểm tra. Nếu tệp file không tồn tại, thông báo tương ứng cũng sẽ được hiển thị.

```
SHfile > $ practice4.sh
1  #!/bin/bash
2
3  echo "Enter the string to check :"
4  read string
5
6  file="test.txt"
7
8  if [ -f "$file" ] ; then
9      if grep -q "$string" "$file"; then
10         echo "String '$string' exist in $file"
11     else
12         echo "String '$string' doesn't exist in $file"
13     fi
14 else
15     echo "$file doesn't exist"
16 fi
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** cat test.txt
ngocanh
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash practice4.sh
Enter the string to check :
ngj
String 'ngj' doesn't exist in test.txt
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash practice4.sh
Enter the string to check :
ngoc
String 'ngoc' exist in test.txt
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** █

Đoạn mã	Giải thích
<code>#!/bin/bash</code>	Dòng đầu tiên xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
<code>echo "Enter the string to check :"</code>	In ra màn hình thông báo yêu cầu người dùng nhập một chuỗi (string) để kiểm tra.
<code>read string</code>	Sử dụng lệnh read để đọc chuỗi (string) mà người dùng nhập từ bàn phím và lưu vào biến string.
<code>file="test.txt"</code>	Gán tên tệp (file) mà chúng ta muốn kiểm tra chuỗi nhập vào biến file.
<code>if [-f "\$file"] ; then</code>	Bắt đầu một câu lệnh điều kiện để kiểm tra xem tệp test.txt có tồn tại không. Nếu tồn tại (-f kiểm tra xem tệp là một tệp thông thường), câu lệnh trong block then sẽ được thực hiện.
<code>if grep -q "\$string" "\$file"; then</code>	Bắt đầu một câu lệnh điều kiện nội bộ để kiểm tra xem chuỗi string có tồn tại trong tệp file không. Sử dụng lệnh grep để tìm kiếm chuỗi trong tệp. Nếu tìm thấy (-q là tùy chọn để chỉ kiểm tra mà không in kết quả), câu lệnh trong block then sẽ được thực hiện.
<code>echo "String '\$string' exist in \$file"</code>	In ra màn hình thông báo cho biết rằng chuỗi string tồn tại trong tệp file.
<code>else</code>	Phần này xảy ra nếu chuỗi string không được tìm thấy trong tệp file.

echo "String '\$string' doesn't exist in \$file"	In ra màn hình thông báo cho biết rằng chuỗi string không tồn tại trong tệp file.
fi	Kết thúc câu lệnh điều kiện nội bộ kiểm tra chuỗi trong tệp file.
else	Phần này xảy ra nếu tệp file không tồn tại.
echo "\$file doesn't exist"	In ra màn hình thông báo cho biết rằng tệp file không tồn tại.
fi	Kết thúc câu lệnh điều kiện kiểm tra tồn tại của tệp file.

2.6. BÀI TẬP ÔN TẬP

1. Tìm hiểu trên Google về việc cài đặt lệnh `git`, sử dụng `git` để tải thư mục ảnh tại đây: https://github.com/locth/OS_LAB2_IMG.git

Viết một file kịch bản để làm những công việc sau:

- Kiểm tra trong thư mục người dùng, nếu thấy thư mục PNG và JPG chưa tồn tại thì tạo 02 thư mục này.
- Di chuyển tất cả file PNG trong thư mục ảnh ở trên vào thư mục PNG. Xuất ra màn hình số lượng ảnh PNG.
- Di chuyển tất cả file JPG trong thư mục ảnh ở trên vào thư mục JPG. Xuất ra màn hình số lượng ảnh JPG.

Trả lời...

a.

- **Cách làm:**

```
#!/bin/bash
```

```
if [ ! -d ~/PNG ]; then
```

```
    mkdir ~/PNG
```

```
    echo "PNG folder has been created."
```

```
fi
```

```
if [ ! -d ~/JPG ]; then
```

```
    mkdir ~/JPG
```

```
    echo "PNG folder has been created."
```

```
fi
```

Kiểm tra xem thư mục PNG đã tồn tại trong thư mục người dùng không:

Dùng điều kiện [! -d ~/PNG] để kiểm tra xem thư mục PNG không tồn tại (điều kiện này sẽ đúng nếu PNG không tồn tại).

Nếu thư mục PNG chưa tồn tại, thì câu lệnh mkdir ~/PNG sẽ tạo thư mục PNG trong thư mục người dùng.

In ra màn hình thông báo "PNG folder has been created." để thông báo rằng thư mục đã được tạo ra.

Tương tự, nó kiểm tra xem thư mục JPG đã tồn tại chưa:

Dùng điều kiện [! -d ~/JPG] để kiểm tra xem thư mục JPG không tồn tại (điều kiện này sẽ đúng nếu JPG không tồn tại).

Nếu thư mục JPG chưa tồn tại, thì câu lệnh mkdir ~/JPG sẽ tạo thư mục JPG trong thư mục người dùng.

In ra màn hình thông báo "PNG folder has been created." để thông báo rằng thư mục đã được tạo ra.

Tóm lại, đoạn mã này kiểm tra xem thư mục PNG và JPG đã tồn tại trong thư mục người dùng không, và nếu chúng chưa tồn tại, nó sẽ tạo các thư mục này và thông báo kết quả.

```
SHfile > $ practice_exercise_a.sh
1  #!/bin/bash
2
3  if [ ! -d ~/PNG ]; then
4      mkdir ~/PNG
5      echo "PNG folder has been created."
6  fi
7  if [ ! -d ~/JPG ]; then
8      mkdir ~/JPG
9      echo "PNG folder has been created."
10 fi
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash practice_exercise_a.sh
PNG folder has been created.
PNG folder has been created.
○ nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$
```

Đoạn mã	Giải thích
#!/bin/bash	Dòng đầu tiên xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
if [! -d ~/PNG]; then	Bắt đầu một câu lệnh điều kiện kiểm tra xem thư mục PNG đã tồn tại trong thư mục người dùng (~) hay chưa. Dấu ! trước -d kiểm tra xem thư mục không tồn tại.

<code>mkdir ~/PNG</code>	Nếu thư mục PNG không tồn tại, sử dụng lệnh <code>mkdir</code> để tạo thư mục PNG trong thư mục người dùng (~).
<code>echo "PNG folder has been created."</code>	In ra màn hình thông báo cho biết rằng thư mục PNG đã được tạo.
<code>fi</code>	Kết thúc câu lệnh điều kiện đầu tiên.
<code>if [! -d ~/JPG]; then</code>	Bắt đầu một câu lệnh điều kiện kiểm tra xem thư mục JPG đã tồn tại trong thư mục người dùng (~) hay chưa.
<code>mkdir ~/JPG</code>	Nếu thư mục JPG không tồn tại, sử dụng lệnh <code>mkdir</code> để tạo thư mục JPG trong thư mục người dùng (~).
<code>echo "PNG folder has been created."</code>	In ra màn hình thông báo cho biết rằng thư mục PNG đã được tạo.
<code>fi</code>	Kết thúc câu lệnh điều kiện thứ hai.

b.

- Cách làm :

```
#!/bin/bash
```

```
find ~/SHfile/OS_LAB2_IMG -name "*.png" -exec mv {} ~/PNG \;
```

```
png_count=$(ls ~/PNG/*.png 2>/dev/null | wc -l)
```

```
echo "number of PNG images moved: $png_count"
```

Báo cáo thực hành môn Hệ điều hành - Giảng viên: Trần Hoàng Lộc.

Sử dụng lệnh find để tìm tất cả các tệp có phần mở rộng .png trong thư mục OS_LAB2_IMG và sau đó di chuyển chúng vào thư mục PNG:

find ~/SHfile/OS_LAB2_IMG -name "*.png" tìm kiếm tất cả các tệp có phần mở rộng .png trong thư mục OS_LAB2_IMG. Kết quả là danh sách các tệp PNG.

-exec mv {} ~/PNG \; sử dụng -exec để di chuyển tệp tìm thấy vào thư mục PNG. {} đại diện cho tệp PNG đã tìm thấy và \; kết thúc lệnh -exec.

Sử dụng lệnh ls để đếm số lượng tệp PNG đã di chuyển vào thư mục PNG:

ls ~/PNG/*.png 2>/dev/null liệt kê tất cả các tệp .png trong thư mục PNG. Dấu 2>/dev/null được sử dụng để ẩn bất kỳ thông báo lỗi nào nếu không có tệp nào tồn tại trong thư mục PNG. Nếu không có tệp PNG nào, lệnh ls sẽ tạo ra một thông báo lỗi, và đoạn mã này ẩn thông báo đó.

| wc -l sử dụng | (pipe) để đưa kết quả đầu ra của lệnh ls vào lệnh wc -l. Lệnh wc -l đếm số dòng trong kết quả đầu ra của lệnh ls, và do đó đếm số lượng tệp PNG đã di chuyển.

In ra màn hình số lượng tệp PNG đã di chuyển:

echo "number of PNG images moved: \$png_count" in ra màn hình số lượng tệp PNG đã di chuyển và lưu trong biến png_count.

Kết quả của đoạn mã này là tìm và di chuyển tất cả các tệp PNG từ thư mục OS_LAB2_IMG vào thư mục PNG, sau đó đếm và in ra màn hình số lượng tệp PNG đã di chuyển.

```
SHfile > $ practice_exercise_b.sh
1  #!/bin/bash
2
3  find ~/SHfile/OS_LAB2_IMG -name "*.png" -exec mv {} ~/PNG \;
4
5  png_count=$(ls ~/PNG/*.png 2>/dev/null | wc -l)
6  echo "number of PNG images moved: $png_count"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** bash practice_exercise_b.sh
number of PNG images moved: 18
- **nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile\$** □

Đoạn mã	Giải thích
#!/bin/bash	Dòng đầu tiên xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
find ~/SHfile/OS_LAB2_IMG -name "*.png" -exec mv { } ~/PNG \;	Sử dụng lệnh find để tìm tất cả các tệp có phần mở rộng .png trong thư mục OS_LAB2_IMG. Sau đó, với mỗi tệp .png tìm thấy, sử dụng -exec để thực hiện lệnh mv để di chuyển tệp đó vào thư mục PNG. {} được sử dụng để đại diện cho tệp .png tìm thấy.
png_count=\$(ls ~/PNG/*.png 2>/dev/null wc -l)	ls ~/PNG/*.png: Lệnh ls được sử dụng để liệt kê tất cả các tệp có phần mở rộng .png

	<p>trong thư mục PNG (thư mục chứa các tệp PNG).</p> <p>2>/dev/null: Đoạn này sử dụng 2>/dev/null để ẩn bất kỳ thông báo lỗi nào được tạo ra bởi lệnh ls. Nếu không có tệp nào tồn tại trong thư mục PNG, lệnh ls sẽ tạo ra một thông báo lỗi. Bằng cách sử dụng 2>/dev/null, chúng ta không in thông báo lỗi đó ra màn hình, mà chỉ quan tâm đến kết quả.</p> <p> : Dấu (pipe) được sử dụng để chuyển kết quả đầu ra của lệnh ls (danh sách các tệp PNG) sang lệnh tiếp theo.</p> <p>wc -l: Lệnh wc -l đếm số dòng trong đầu vào mà nó nhận được. Trong trường hợp này, nó đếm số dòng trong danh sách các tệp PNG từ lệnh ls.</p> <p>png_count=: Kết quả của lệnh wc -l là số lượng dòng (tệp PNG) trong danh sách. Số này được gán vào biến png_count.</p>
echo "number of PNG images moved: \$png_count"	Sử dụng echo để in ra màn hình thông báo về số lượng tệp hình ảnh PNG đã được di chuyển và lưu trong biến png_count.

c.

- Cách làm :

Báo cáo thực hành môn Hệ điều hành - Giảng viên: Trần Hoàng Lộc.

```
#!/bin/bash
```

```
find ~/SHfile/OS_LAB2_IMG -name "*.jpg" -exec mv {} ~/JPG \;
```

```
jpg_count=$(ls ~/JPG/*.jpg 2>/dev/null | wc -l)
```

```
echo "number of JPG images moved: $jpg_count"
```

Sử dụng lệnh find để tìm tất cả các tệp có phần mở rộng .png trong thư mục OS_LAB2_IMG và sau đó di chuyển chúng vào thư mục PNG:

find ~/SHfile/OS_LAB2_IMG -name "*.jpg" tìm kiếm tất cả các tệp có phần mở rộng .png trong thư mục OS_LAB2_IMG. Kết quả là danh sách các tệp JPG.

-exec mv {} ~/JPG \; sử dụng -exec để di chuyển tệp tìm thấy vào thư mục JPG. {} đại diện cho tệp JPG đã tìm thấy và \; kết thúc lệnh -exec.

Sử dụng lệnh ls để đếm số lượng tệp JPG đã di chuyển vào thư mục JPG:

ls ~/JPG/*.jpg 2>/dev/null liệt kê tất cả các tệp .png trong thư mục JPG. Dấu 2>/dev/null được sử dụng để ẩn bất kỳ thông báo lỗi nào nếu không có tệp nào tồn tại trong thư mục JPG. Nếu không có tệp JPG nào, lệnh ls sẽ tạo ra một thông báo lỗi, và đoạn mã này ẩn thông báo đó.

| wc -l sử dụng | (pipe) để đưa kết quả đầu ra của lệnh ls vào lệnh wc -l. Lệnh wc -l đếm số dòng trong kết quả đầu ra của lệnh ls, và do đó đếm số lượng tệp JPG đã di chuyển.

In ra màn hình số lượng tệp JPG đã di chuyển:

echo "number of JPG images moved: \$jpg_count" in ra màn hình số lượng tệp JPG đã di chuyển và lưu trong biến jpg_count.

Kết quả của đoạn mã này là tìm và di chuyển tất cả các tệp JPG từ thư mục OS_LAB2_IMG vào thư mục JPG, sau đó đếm và in ra màn hình số lượng tệp JPG đã di chuyển.


```
SHfile > $ practice_exercise_c.sh
1  #!/bin/bash
2
3  find ~/SHfile/OS_LAB2_IMG -name "*.jpg" -exec mv {} ~/JPG \;
4
5  jpg_count=$(ls ~/JPG/*.jpg 2>/dev/null | wc -l)
6  echo "number of JPG images moved: $jpg_count"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● **nguyennguyennngocanh-22520058@AndreNguyen**:~/SHfile\$ bash practice_exercise_c.sh
number of JPG images moved: 60

Đoạn mã	Giải thích
#!/bin/bash	Dòng đầu tiên xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
find ~/SHfile/OS_LAB2_IMG -name "*.jpg" -exec mv {} ~/JPG \;	Sử dụng lệnh find để tìm tất cả các tệp có phần mở rộng .jpg trong thư mục OS_LAB2_IMG. Sau đó, với mỗi tệp .jpg tìm thấy, sử dụng -exec để thực hiện lệnh mv để di chuyển tệp đó vào thư mục JPG. {} được sử dụng để đại diện cho tệp .jpg tìm thấy.

<pre>jpg_count=\$(ls ~/JPG/*.png 2>/dev/null wc -l)</pre>	<p>ls ~/JPG/*.jpg: Lệnh ls được sử dụng để liệt kê tất cả các tệp có phần mở rộng .jpg trong thư mục JPG (thư mục chứa các tệp JPG).</p> <p>2>/dev/null: Đoạn này sử dụng 2>/dev/null để ẩn bất kỳ thông báo lỗi nào được tạo ra bởi lệnh ls. Nếu không có tệp nào tồn tại trong thư mục JPG, lệnh ls sẽ tạo ra một thông báo lỗi. Bằng cách sử dụng 2>/dev/null, chúng ta không in thông báo lỗi đó ra màn hình, mà chỉ quan tâm đến kết quả.</p> <p> : Dấu (pipe) được sử dụng để chuyển kết quả đầu ra của lệnh ls (danh sách các tệp JPG) sang lệnh tiếp theo.</p> <p>wc -l: Lệnh wc -l đếm số dòng trong đầu vào mà nó nhận được. Trong trường hợp này, nó đếm số dòng trong danh sách các tệp JPG từ lệnh ls.</p> <p>jpg_count=: Kết quả của lệnh wc -l là số lượng dòng (tệp JPG) trong danh sách. Số này được gán vào biến jpg_count.</p>
<pre>echo "number of JPG images moved: \$jpg_count"</pre>	<p>Sử dụng echo để in ra màn hình thông báo về số lượng tệp hình ảnh JPG đã được di chuyển và lưu trong biến jpg_count.</p>

2. Tạo ra một file text tên monhoc .txt chứa danh sách mã môn học của sinh viên trong học kỳ này, mỗi mã nằm trên một dòng. Viết một file kịch bản thực hiện các việc sau:

a. Yêu cầu người dùng nhập vào họ và tên (không dấu), tạo ra thư mục có tên tương ứng với thông tin người dùng vừa nhập

b. Đọc file text monhoc .txt ở trên, ở trong thư mục vừa tạo ở câu a, với mỗi môn học, tạo ra một thư mục có tên tương ứng với mã môn đó.

Trả lời

a.

- **Cách làm:**

```
#!/bin/bash
```

```
echo "Enter your full name (without accent marks): "
```

```
read full_name
```

```
directory_name="${full_name// /_}"
```

```
mkdir "$directory_name"
```

```
echo "Directory '$directory_name' has been created."
```

Yêu cầu người dùng nhập họ và tên (không dấu):

```
echo "Enter your full name (without accent marks): "
```

```
read full_name
```

Tạo biến directory_name để lưu tên thư mục mới. Để tạo tên thư mục, chúng ta thay thế khoảng trắng bằng gạch dưới _ (để đảm bảo tên thư mục hợp lệ):

```
directory_name="${full_name// /_}"
```

Ví dụ: Nếu người dùng nhập "John Doe," thì biến directory_name sẽ chứa "John_Doe."

Sử dụng lệnh mkdir để tạo thư mục mới với tên được lưu trong biến directory_name:

```
mkdir "$directory_name"
```

Báo cáo thực hành môn Hệ điều hành - Giảng viên: Trần Hoàng Lộc.

In ra màn hình thông báo cho biết rằng thư mục đã được tạo ra:

```
echo "Directory '$directory_name' has been created."
```

Kết quả của đoạn mã này là tạo một thư mục mới có tên dựa trên họ và tên của người dùng (không dấu) và thông báo rằng thư mục đã được tạo thành công.

```
SHfile > $ practice_exercise_2a.sh
1  #!/bin/bash
2
3  echo "Enter your full name (without accent marks): "
4  read full_name
5
6  directory_name="${full_name// /_}"
7  mkdir "$directory_name"
8
9  echo "Directory '$directory_name' has been created."
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash practice_exercise_2a.sh
Enter your full name (without accent marks):
Nguyen Nguyen Ngoc Anh
Directory 'Nguyen_Nguyen_Ngoc_Anh' has been created.
○ nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$
```

Đoạn mã	Giải thích
<code>#!/bin/bash</code>	Dòng đầu tiên xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
<code>echo "Enter your full name (without accent marks):"</code>	In ra màn hình yêu cầu người dùng nhập họ và tên (không dấu).
<code>read full_name</code>	Sử dụng lệnh <code>read</code> để đọc họ và tên (full name) mà người dùng nhập từ bàn phím và lưu vào biến <code>full_name</code> .
<code>directory_name="\${full_name// /_}"</code>	Tạo biến <code>directory_name</code> bằng cách thay thế khoảng trắng bằng gạch dưới <code>_</code> trong biến <code>full_name</code> .
<code>mkdir "\$directory_name"</code>	Sử dụng lệnh <code>mkdir</code> để tạo thư mục mới với tên là <code>directory_name</code> .
<code>echo "Directory '\$directory_name' has been created."</code>	In ra màn hình thông báo cho biết rằng thư mục đã được tạo ra.

b.

- Cách làm:

```
#!/bin/bash
```

```
while IFS= read -r subject_code
do
    subject_directory="Nguyen_Nguyen_Ngoc_Anh/$subject_code"
    mkdir "$subject_directory"
    echo "Directory '$subject_directory' has been created."
done < monhoc.txt
```

Người dùng được yêu cầu nhập họ và tên (không dấu).

Một thư mục gốc được tạo ra với tên dựa trên họ và tên của người dùng, và khoảng trắng trong tên được thay thế bằng gạch dưới _.

Danh sách mã môn học được đọc từ tệp monhoc.txt dựa trên một vòng lặp while. Mỗi mã môn học được lưu vào biến subject_code.

Thư mục con với tên tương ứng với mã môn học được tạo trong thư mục gốc đã được tạo trước đó.

Kết quả của đoạn mã này là thư mục gốc với tên của người dùng (không dấu) và bên trong nó, một thư mục con được tạo cho mỗi mã môn học từ tệp monhoc.txt.

```
SHfile > $ practice_exercise_2b.sh
```

```
1  #!/bin/bash
2
3  while IFS= read -r subject_code
4  do
5      subject_directory="Nguyen_Nguyen_Ngoc_Anh/$subject_code"
6      mkdir "$subject_directory"
7      echo "Directory '$subject_directory' has been created."
8  done < monhoc.txt
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$ touch practice_exercise_2b.sh
● nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$ chmod +x practice_exercise_2b.sh
● nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash practice_exercise_2b.sh
practice_exercise_2b.sh: line 3: -r: command not found
● nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$ bash practice_exercise_2b.sh
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT007.011 - VN' has been created.
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT007.011.1 - VN(HT1)' has been created.
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT005.018 - VN' has been created.
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT005.018.1 - VN(HT1)' has been created.
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT008.014 - VN' has been created.
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT008.014.1 - VN(HT1)' has been created.
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT004.0117 - VN' has been created.
Directory 'Nguyen_Nguyen_Ngoc_Anh/IT004.0117.1 - VN(HT1)' has been created.
○ nguyennguyenngocanh-22520058@AndreNguyen:~/SHfile$
```

Đoạn mã	Giải thích
<code>#!/bin/bash</code>	Dòng đầu tiên xác định loại shell mà script sử dụng, trong trường hợp này là Bash.
<code>while IFS= read -r subject_code; do</code>	Bắt đầu một vòng lặp while để đọc từng dòng từ tệp monhoc.txt. IFS= được sử dụng để không loại bỏ khoảng trắng khi đọc. -r được sử dụng để đọc các dòng "raw" (không xử lý ký tự đặc biệt).
<code>subject_directory= "Nguyen_Nguyen_Ngoc_Anh/\$subject_code"</code>	Tạo biến subject_directory chứa đường dẫn đầy đủ đến thư mục con với tên tương ứng với mã môn học.
<code>mkdir "\$subject_directory"</code>	Sử dụng lệnh mkdir để tạo thư mục con với đường dẫn trong biến subject_directory.
<code>echo "Directory '\$subject_directory' has been created."</code>	In ra màn hình thông báo cho biết rằng thư mục con đã được tạo.
<code>done < monhoc.txt</code>	Kết thúc vòng lặp while và sử dụng < để đọc dữ liệu từ tệp monhoc.txt cho đến khi tệp này được đọc hết.