

```
# ! /bin/bash
echo "Nhap do n: "
read n
if ! [[ "$n" =~ ^[0-9]+$ ]]; then
    echo "Vui lòng nhập một số nguyên dương."
    exit 1
fi
```

```
is_prime() {
    local num=$1
    if [ $num -lt 2 ]; then return 1; fi
    for ((i=2; i*i <= num; i++)); do
        if [ $(($num % i)) -eq 0 ]; then
            return 1; fi
        done
    return 0
}
```

```
is_square() {
    local num=$1
    local sqrt=$(echo "sqrt($num)" | bc)
    local squared=$(echo "$sqrt*$sqrt" | bc)
    if [ "$squared" -eq "$num" ]; then return 0;
    else return 1; fi
}

if is_prime $n; then
    echo "$n là số nguyên tố."
else if is_square $n; then
    echo "$n là số chính phương."
else
    echo "$n không phải số nguyên tố hoặc số chính phương."
fi
```

3) #!/bin/bash

```
function IntorchangeSort() {  
    local n=${#a[@]}  
    for ((i=0; i<n-1; i++)), do  
        for ((j=i+1; j<n; j++)), do  
            if ((a[i] > a[j])), then  
                temp=${a[j]}  
                a[j]= ${a[i]}  
                a[i]= $temp  
            fi  
        done  
    done  
}
```

```
declare -a a  
echo -n "N: "  
read n  
for ((i=0; i<n; i++)), do  
    echo -n "a[i] = "  
    read a[i]  
done
```

```
echo "Before ${a[*]}"  
IntorchangeSort a  
echo "After ${a[*]}"
```

4) #!/bin/bash

```
read -p "File: " filename  
touch "$filename"
```

echo "nhập nội dung cho file (gõ 'finished' để kết)"

```
while true, do  
    read line  
    if ["$line" = "finished"], then  
        break,  
    fi  
    echo "$line" >> "$filename"  
done
```

MÔN HỆ ĐIỀU HÀNH MÃ NGUỒN MỞ

TUẦN 11

LẬP TRÌNH SHELL

Bài 1: Viết 1 chương trình Shell để cho người dùng nhập vào 1 số, kiểm tra số đó có phải là số nguyên tố, hoặc số đó là số chính phương? Tương tự, hãy sử dụng đối số mà không cần để người dùng nhập.

Bài 2: Viết 1 chương trình Shell, cho người dùng nhập vào 1 số n , xuất ra các số chính phương và số nguyên tố nằm trong khoảng 1 đến n . Tương tự, hãy sử dụng đối số mà không cần để người dùng nhập.

Bài 3: Viết 1 chương trình Shell, cho người dùng nhập vào 1 mảng, sắp xếp mảng theo giá trị tăng dần. Tương tự, hãy sử dụng đối số mà không cần để người dùng nhập.

Lưu ý: khai báo và cho người dùng nhập chiều dài của mảng, dùng vòng lặp for để nhập vào giá trị từng phần tử trong mảng.

Bài 4: Viết 1 chương trình Shell, cho người dùng nhập vào 1 tên file, tạo ra file với tên vừa nhập, sau đó cho người dùng nhập nội dung của file đó vào.

Lưu ý: để nhập nội dung của 1 file vào, có thể dùng 1 vòng lặp, kiểm tra điều kiện xem người dùng có nhập chữ finished không? Nếu nhập chữ finished thì kết thúc vòng lặp. Nếu không thì dùng ký tự chuyển hướng để nhập vào file.

Ví dụ:

```
while true
```

```
do
```

```
    read a
```

```
    if [ a == "finished" ]; then
```

```
        break
```

```
    cat $a > file
```

```
done
```

Bài 5: từ file đã làm ở bài 4, hãy sắp xếp thứ tự của nội dung của file trên.

Lưu ý: ta có thể sử dụng sắp xếp bằng câu lệnh sort.

Ví dụ:

```
string.txt
```

```
qwerty
qaz
$ sort string.txt
qaz
qwerty
```

Sắp xếp theo số:

```
number.txt
1000
0.1
1.23
-1
-100
$ sort -n number.txt
-100
-1
0.1
1.23
1000
```

Bài 5: Viết shell script có dạng: fileread.sh file1 file2 file3 (fileread.sh là tên shell script, file1, file2 là tên các đối số)

Chức năng:

Đọc một dòng từ bàn phím và ghi lên file theo cách sau:

- Vào cuối file1 nếu dòng chứa ít nhất một chữ cái
- Vào cuối file2 nếu dòng chứa ít nhất một chữ số và không chứa bất kỳ chữ cái nào
- Ghi vào file3 trong trường hợp còn lại

Kiểm tra các đối số, nếu khác 3 đối số, hiển thị thông báo: “command: fileread.sh file1 file2 file3”

Thống kê có bao nhiêu ký tự trong cả 3 file

Đối với file1, ghi tất cả ký tự trên cùng 1 dòng, có khoảng trắng giữa các ký tự

Lưu ý: cho người dùng nhập vào 1 biến, biến này sẽ lưu dữ liệu của 1 dòng. Cho vòng lặp chạy trong các từ trong biến vừa nhập, kiểm tra nó có phải chữ số hay chữ cái.

Ví dụ:

```

# Nhap input tu prompt
read -p "- Nhap vao 1 chuoi bat ki: " STRING

if [[ -z ${STRING} ]];then
    echo "-+ Chuoi nhap vao rong."
    echo "-+ Exit."
    exit 1
fi

# Dem so luong tu voi command 'wc -w'
number_word=$(echo ${STRING} | wc -w)
echo "- So luong tu: ${number_word}"

# In tung tu tren tung dong
for word in $STRING
do
    echo ${word}
done

exit 0

```

Kiểm tra 1 từ có phải là số?

```

if [[ $((foo)) != $foo ]]; then
    echo "Not just a number!"
fi

```

Bài 1: Viết 1 chương trình Shell để cho người dùng nhập vào 1 số, kiểm tra số đó có phải là số nguyên tố, hoặc số đó là số chính phương? Tương tự, hãy sử dụng đối số mà không cần để người dùng nhập.

```
1 #!/bin/bash
2
3 function SNT() {
4     local n=$1
5     if ((n<2)); then
6         return 0
7     else
8         for ((i=2; i<=((n/2)); i++)); do
9             if ((n%i==0)); then
10                 return 0
11             fi
12         done
13         return 1
14     fi
15 }
16
17 function SCP() {
18     local n=$1
19     local sqrt=$(echo "scale=0; sqrt($n)" | bc)
20     if ((sqrt*sqrt==n)); then
21         return 1
22     fi
23     return 0
24 }
25
26 echo -n "N: "
27 read n
28 echo -n "SNT: "
29 SNT $n
30 if [ $? -eq 1 ]; then
31     echo "True"
32 else
33     echo "False"
34 fi
35
36 echo -n "SCP: "
37 SCP $n
38 if [ $? -eq 1 ]; then
39     echo "True"
40 else
41     echo "False"
42 fi
```

```
1 #!/bin/bash
2
3 function SNT() {
4     local n=$1
5     if ((n<2)); then
6         return 0
7     else
8         for ((i=2; i<=((n/2)); i++)); do
9             if ((n%i==0)); then
10                 return 0
11             fi
12         done
13         return 1
14     fi
15 }
16
17 function SCP() {
18     local n=$1
19     local sqrt=$(echo "scale=0; sqrt($n)" | bc)
20     if ((sqrt*sqrt==n)); then
21         return 1
22     fi
23     return 0
24 }
25
26 n=$1
27 echo -n "SNT: "
28 SNT $n
29 if [ $? -eq 1 ]; then
30     echo "True"
31 else
32     echo "False"
33 fi
34
35 echo -n "SCP: "
36 SCP $n
37 if [ $? -eq 1 ]; then
38     echo "True"
39 else
40     echo "False"
41 fi
```

Bài 2: Viết 1 chương trình Shell, cho người dùng nhập vào 1 số n, xuất ra các số chính phương và số nguyên tố nằm trong khoảng 1 đến n. Tương tự, hãy sử dụng đối số mà không cần để người dùng nhập.

```
1 #!/bin/bash
2
3 function SNT() {
4     local n=$1
5     if ((n<2)); then
6         return 0
7     else
8         for ((i=2; i<=((n/2)); i++)); do
9             if ((n%i==0)); then
10                 return 0
11             fi
12         done
13         return 1
14     fi
15 }
16
17 function SCP() {
18     local n=$1
19     local sqrt=$(echo "scale=0; sqrt($n)" | bc)
20     if ((sqrt*sqrt==n)); then
21         return 1
22     fi
23     return 0
24 }
25
26 echo -n "N: "
27 read n
28
29 snt=""
30 scp=""
31 for ((j=1; j<=n; j++)); do
32     SNT $j
33     if (($?=1)); then
34         snt="$snt $j"
35     fi
36     SCP $j
37     if (($?=1)); then
38         scp="$scp $j"
39     fi
40 done
41
42 echo "SNT: $snt"
43 echo "SCP: $scp"
44
```

```
1 #!/bin/bash
2
3 function SNT() {
4     local n=$1
5     if ((n<2)); then
6         return 0
7     else
8         for ((i=2; i<=((n/2)); i++)); do
9             if ((n%i==0)); then
10                 return 0
11             fi
12         done
13         return 1
14     fi
15 }
16
17 function SCP() {
18     local n=$1
19     local sqrt=$(echo "scale=0; sqrt($n)" | bc)
20     if ((sqrt*sqrt==n)); then
21         return 1
22     fi
23     return 0
24 }
25
26 n=$1
27
28 snt=""
29 scp=""
30 for ((j=1; j<=n; j++)); do
31     SNT $j
32     if (($?=1)); then
33         snt="$snt $j"
34     fi
35     SCP $j
36     if (($?=1)); then
37         scp="$scp $j"
38     fi
39 done
40
41 echo "SNT: $snt"
42 echo "SCP: $scp"
43
```

Bài 3: Viết 1 chương trình Shell, cho người dùng nhập vào 1 mảng, sắp xếp mảng theo giá trị tăng dần. Tương tự, hãy sử dụng đối số mà không cần để người dùng nhập.

```
1  #!/bin/bash
2
3  function InterchangeSort() {
4      local n=${#a[@]}
5      for (( i=0; i<n-1; i++ )); do
6          for (( j=i+1; j<n; j++ )); do
7              if (( a[i] > a[j] )); then
8                  temp=${a[i]}
9                  a[i]=${a[j]}
10                 a[j]=$temp
11             fi
12         done
13     done
14 }
15
16 declare -a a
17 echo -n "N: "
18 read n
19
20 for (( i=0; i<n; i++ )); do
21     echo -n "a[$i] = "
22     read a[i]
23 done
24
25 echo "Before: ${a[*]}"
26 InterchangeSort a
27 echo "After: ${a[*]}"
```

```
1  #!/bin/bash
2
3  function InterchangeSort() {
4      local n=${#a[@]}
5      for (( i=0; i<n-1; i++ )); do
6          for (( j=i+1; j<n; j++ )); do
7              if (( a[i] > a[j] )); then
8                  temp=${a[i]}
9                  a[i]=${a[j]}
10                 a[j]=$temp
11             fi
12         done
13     done
14 }
15
16 n=$1
17 # Lấy các phần tử của mảng từ đối số thứ 2 trở đi
18 declare -a a=("${@:2}")
19
20 echo "Before: ${a[*]}"
21 InterchangeSort a
22 echo "After: ${a[*]}"
23
```