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.

* Nhập số

*#!bin/bash*

checkPrime() {

**for** ((i=2;i<$1;i++)); **do**

**if** [ $[$1%$i] -eq 0 ]; **then**

**return** 1

**fi**

**done**

**return** 0

}

ktSoChinhPhuong() {

**for** ((i=1;i<=$1;i++)); **do**

**if** [ $[$i\*$i] -eq $1 ]; **then**

**return** 0

**fi**

**done**

**return** 1

}

echo "Nhap mot so"; read n

echo "Ban da nhap so $n"

*# xử lí số nguyên tố*

**if** [ $n -gt 2 ]; **then**

**if** $(checkPrime $n); **then**

        echo "$n la phai so nguyen to"

**else**

        echo "$n khong phai so nguyen to"

**fi**

**fi**

*# xử lí số chính phương*

**if** [ $n -lt 2 ]; **then**

    echo "$n khong phai so chinh phuong!"

    exit 1

**else**

**if** $(ktSoChinhPhuong $n); **then**

        echo "$n la phai so chinh phuong"

**else**

        echo "$n khong phai chinh phuong"

**fi**

**fi**

* Đối số
* *#!bin/bash*
* checkPrime() {
* **for** ((i=2;i<$1;i++)); **do**
* **if** [ $[$1%$i] -eq 0 ]; **then**
* **return** 1
* **fi**
* **done**
* **return** 0
* }
* ktSoChinhPhuong() {
* **for** ((i=1;i<=$1;i++)); **do**
* **if** [ $[$i\*$i] -eq $1 ]; **then**
* **return** 0
* **fi**
* **done**
* **return** 1
* }
* *# xử lí số nguyên tố*
* **if** [ $1 -gt 2 ]; **then**
* **if** [ $(checkPrime $1) ]; **then**
* echo "$1 la phai so nguyen to"
* **else**
* echo "$1 khong phai so nguyen to"
* **fi**
* **fi**
* *# xử lí số chính phương*
* **if** [ $1 -lt 2 ]; **then**
* echo "$1 khong phai so chinh phuong!"
* exit 1
* **else**
* **if** [ $(ktSoChinhPhuong $1) ]; **then**
* echo "$1 la phai so chinh phuong"
* **else**
* echo "$1 khong phai chinh phuong"
* **fi**
* **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.

* Nhập số
* *#!bin/bash*
* checkPrime() {
* **for** ((i=2;i<$1;i++)); **do**
* **if** [ $[$1%$i] -eq 0 ]; **then**
* **return** 1
* **fi**
* **done**
* **return** 0
* }
* ktSoChinhPhuong() {
* **for** ((i=1;i<=$1;i++)); **do**
* **if** [ $[$i\*$i] -eq $1 ]; **then**
* **return** 0
* **fi**
* **done**
* **return** 1
* }
* echo "Nhap mot so"; read n
* echo "Ban da nhap so $n"
* echo "Cac so nguyen to tu 1 den $n :"
* *# xử lí số nguyên tố*
* **for** ((i=1;i<=$n;i++)); **do**
* **if** $(checkPrime $i) ; **then**
* echo $i
* **fi**
* **done**
* echo "Cac so chinh phuong tu 1 den $n"
* *# xử lí số chính phương*
* **for** ((i=1;i<=$n;i++)); **do**
* **if** $(ktSoChinhPhuong $i)  ; **then**
* echo $i
* **fi**
* **done**
* Đối số
* *#!bin/bash*
* checkPrime() {
* **for** ((i=2;i<$1;i++)); **do**
* **if** [ $[$1%$i] -eq 0 ]; **then**
* **return** 1
* **fi**
* **done**
* **return** 0
* }
* ktSoChinhPhuong() {
* **for** ((i=1;i<=$1;i++)); **do**
* **if** [ $[$i\*$i] -eq $1 ]; **then**
* **return** 0
* **fi**
* **done**
* **return** 1
* }
* echo "Nhap mot so"; read n
* echo "Ban da nhap so $n"
* echo "Cac so nguyen to tu 1 den $n :"
* *# xử lí số nguyên tố*
* **for** ((i=1;i<=$n;i++)); **do**
* **if** $(checkPrime $i) ; **then**
* echo $i
* **fi**
* **done**
* echo "Cac so chinh phuong tu 1 den $n"
* *# xử lí số chính phương*
* **for** ((i=1;i<=$n;i++)); **do**
* **if** $(ktSoChinhPhuong $i)  ; **then**
* echo $i
* **fi**
* **done**

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.

* Nhập số
* *#!/bin/bash*
* echo "Nhap so luong phan tu: "; read n
* echo "Nhap phan tu cua mang!"
* **for** ((i=0;i<$n;i++)); **do**
* read a[$i]
* **done**
* **for** ((i=0;i<$n;i++)); **do**
* **for** ((j=$i+1;j<$n-1;j++)); **do**
* **if** [ ${a[$i]} -gt ${a[$j]} ]; **then**
* temp=${a[$i]}
* a[$i]=${a[$j]}
* a[$j]=$temp
* **fi**
* **done**
* **done**
* echo "Mang sau khi duoc sap xep lai ${a[@]}"
* exit 0
* Đối số
* *#!/bin/bash*
* n=($#)
* a=($\*)
* **for** ((i=0;i<$n;i++)); **do**
* **for** ((j=$i+1;j<$n-1;j++)); **do**
* **if** [ ${a[$i]} -gt ${a[$j]} ]; **then**
* temp=${a[$i]}
* a[$i]=${a[$j]}
* a[$j]=$temp
* **fi**
* **done**
* **done**
* echo "Mang sau khi duoc sap xep lai ${a[@]}"
* exit 0

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.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

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:

1. Vào cuối file1 nếu dòng chứa ít nhất một chữ cái
2. Vào cuối file2 nếu dòng chứa ít nhất một chữ số và không chứa bất kỳ chữ cái nào
3. 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:

Text

Description automatically generated

Kiểm tra 1 từ có phải là số?

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

fi

Text

Description automatically generated