**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO MÔN HỌC**

**PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ**

**Nhóm 2**

Phạm Ngọc Hòa – DTH185278

Trần Tấn Duy– DTH185250

Lê Ngọc Thắng– DTH185388

Dương Khởi Minh– DTH185488

Mai Quang Huy– DTH185258

Trần Hoàng Toàn– DTH185406

|  |  |
| --- | --- |
|  |  |

**AN GIANG, THÁNG 5 NĂM 2022**

**Danh sách thành viên và nội dung phân công**

**Nhóm 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ Tên** | **MSSV** | **Công Việc** |
| 1 | Phạm Ngọc Hòa | DTH185278 | Làm buổi 3, ôn các buổi còn lại |
| 2 | Trần Tấn Duy | DTH185250 | Làm buổi 2, ôn các buổi còn lại |
| 3 | Lê Ngọc Thắng | DTH185388 | Làm buổi 4, ôn các buổi còn lại |
| 4 | Dương Khởi Minh | DTH185488 | Làm buổi 2, ôn các buổi còn lại |
| 5 | Mai Quang Huy | DTH185258 | Làm buổi 4, ôn các buổi còn lại |
| 6 | Trần Hoàng Toàn | DTH185406 | Làm buổi 3, ôn các buổi còn lại |

# **BUỔI 2**

1. **Script cơ bản**
2. **Viết script nhập vào 2 số (dạng đối số dòng lệnh) và tính tổng**

#!/bin/sh

# Bai 1: Tinh tong

echo $1 + $2 = `expr $1 + $2`

Text

Description automatically generated

1. **Viết script thực hiện các phép toán cơ bản (+, -, x, /) được nhập từ dòng lệnh**

#!/bin/bash

#Bai 2: Phep toan

if test $#=3

then

case $2 in

+) let z=$1+$3;;

-) let z=$1-$3;;

/) let z=$1/$3;;

x|X) let z=$1\*$3;;

\*) echo $2 khong hop le. Dung +, -, x, /.

exit;;

esac

echo $1 $2 $3 = $z

fi

Text

Description automatically generated

1. **Viết script tìm số lớn nhất trong 3 số được nhập từ dòng lệnh**

#bai 3

declare -a a

a=( [0]=$1 [1]=$2 [2]=$3 )

max=${a[0]}

l=${#a[\*]}

for ((i=0;i<$l;i++))

do

if [ $max -le ${a[i]} ];

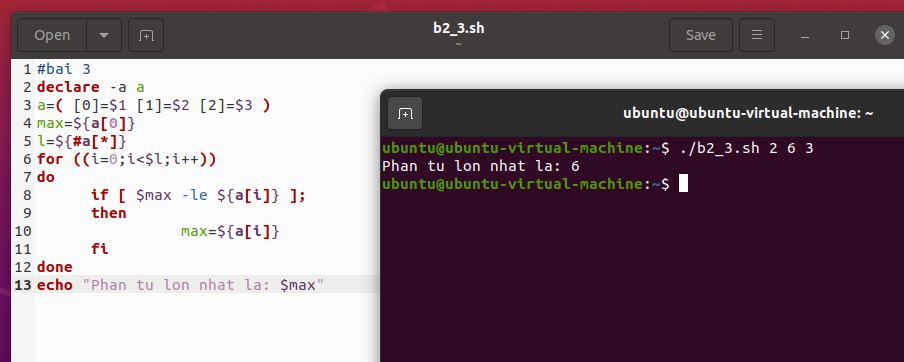
then

max=${a[i]}

fi

done

echo "Phan tu lon nhat la: $max"



1. **IN bảng nhân của một số nhập vào n (in bảng cửu chương)**

#bai 4

echo -n "Ban muon in bang cuu chuong may: "

read so

for ((i=1;i<=10;i++))

do

echo "$so x $i = `expr $so \\* $i`"

done

Text

Description automatically generated

1. **Tính tong: S= 1-2+3-4+5…n.**

#Bai 5:

echo -n "Nhap n: "

read n

if [ $n == 0 ]

then

echo "Ket qua = 0"

else if [ $n == 1 ]

then

echo "Ket qua = 1"

else if [ $n < 0 ]

then

echo "N khong hop le"

else

tong = 0

for ((i=2; i<= $n; i++))

do

if [ i % 2 == 1 ]

then

tong = $(($tong - $i))

else

tong = $(($tong + $i))

fi

done

echo "Ket qua = " $tong

fi

A screenshot of a computer

Description automatically generated

1. **Viết script tính tổng các ký số của một số được nhập vào**

#Bai 6: Tinh tong so duoc nhap

tong=0

temp=$1

while [ $temp -ne 0 ]

do

tong=$(($tong+($temp%10)))

temp=$(($temp/10))

done

echo "Ket qua la: $tong"

A screenshot of a computer

Description automatically generated with medium confidence

1. **Tính giai thừa của một số**

#Bai 7: giai thua

echo -n "Nhap n: "

read n

sum=1;

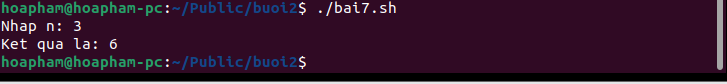
for ((i=1;i<=n;i++))

do

sum=$(($sum \* $i))

done

echo "Ket qua la: $sum"



1. **Viết script để in ra**

#Bai 8

i = 1

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

do

for ((n=1;n<=i;n++))

do

echo -n "\*"

done

echo " "

done

echo ""

i=1

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

do

for ((n=1;n<=i;n++))

do

echo -n $i

done

echo " "

done

echo ""

i=1

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

do

for ((n=1;n<=i;n++))

do

echo -n $n

done

echo " "

done

Text

Description automatically generated

1. **Tạo menu tương tác với người dùng**

**(Hiển thị ngày giờ, thư mục hiện hành người dùng và lịch )**

#!/bin/bash

echo "---------------------------------------"

echo "Main Menu"

echo "---------------------------------------"

echo "[1] Show today date/time"

echo "[2] Show all files in current directory"

echo "[3] Show users"

echo "[4] Show calendar"

echo "[5] Exit/Stop"

echo "---------------------------------------"

echo "Nhap so:"

read so

if [ $so -eq 1 ]

then

echo `date +%c`

elif [ $so -eq 2 ]

then

echo `pwd`

elif [ $so -eq 3 ]

then

echo `whoami`

elif [ $so -eq 4 ]

then

echo `ncal`

elif [ $so -eq 5 ]

then

exit

else

echo "So khong co trong Menu!"

exit

fi

Text

Description automatically generated

1. **Tìm số lớn nhất trong dãy số bất kỳ truyền vào từ dòng lệnh (dùng hàm max/min 2 số để tìm max/min dãy số bất kỳ)**

#!/bin/bash

if [ $# -lt 1 ]; then

echo "Phai nhap it nhat 1 so."

else

max=$1

min=$1

for arg

do

if [ $max -lt $arg ]; then

max=$arg

fi

if [ $min -gt $arg ]; then

min=$arg

fi

done

echo "Max = $max"

echo "Min = $min"

fi

A screenshot of a computer

Description automatically generated

1. **In ra các số nguyên tố trong dãy số bất kỳ truyền vào từ dòng lệnh (dùng hàm)**

#!/bin/bash

is\_prime(){

n=$1

if [ $n -lt 1 ];then

return 0

fi

for ((i=2;i<n;i++));do

let "k=$n%$i"

if [ $k -eq 0 ];then

return 0

fi

done

return 1

}

echo -n "So nguyen to: "

for arg

do

is\_prime $arg

if [ $? -ne 0 ];then

echo -n "$arg "

fi

done

echo ""

A screenshot of a computer

Description automatically generated

1. **Nhập vào 1 mảng và sắp xếp mảng tăng dần**

#Bai 15: UCLN mang

#!/bin/bash

echo -n "Nhap n: "

read n

declare -a arr

for ((i=0;i<n;i++))

do

echo -n "Nhap phan tu thu `expr $i + 1`: "

read arr[$i]

done

for ((i = 0; i<n; i++))

do

for((j = 0; j<n-i-1; j++))

do

if [ ${arr[j]} -gt ${arr[$((j+1))]} ]

then

# swap

temp=${arr[j]}

arr[$j]=${arr[$((j+1))]}

arr[$((j+1))]=$temp

fi

done

done

echo -n "Mang sau khi sap xep tang dan: "

echo "${arr[\*]}"

A screenshot of a computer

Description automatically generated with medium confidence

1. **Nhập 1 mảng n số nguyên (nhập từ bàn phím). In mảng ngược**

#!/bin/bash

declare -a a

echo -n "Nhap n: "

read n

for ((i=1;i<=n;i++));do

echo -n "a[$i]= "

read m

a[$i]=$m

done

echo -n "Mang chua sap xep: "

echo ${a[\*]}

for ((i=1;i<=n-1;i++));do

for ((j=i+1;j<=n;j++));do

tam=${a[$i]}

a[$i]=${a[$j]}

a[$j]=$tam

done

done

echo ${a[\*]}

Graphical user interface, text

Description automatically generated

1. **In ra các phần tử chẵn lẻ,Tính tổng các phần tử trong mảng. (dùng hàm tổng 2 số)**

#Bai 14:

#!/bin/bash

echo -n "Nhap n: "

read n

declare -a mang

for ((i=0;i<n;i++))

do

echo -n "Nhap phan tu thu `expr $i + 1`: "

read mang[$i]

done

echo ""

echo -n "Cac phan tu chan la: "

for ((i=0;i<n;i++))

do

arrI=${mang[i]}

if [ `expr $arrI % 2` -eq 0 ]; then

echo -n "${mang[i]} "

fi

done

echo ""

echo -n "Cac phan tu le la: "

for ((i=0;i<n;i++))

do

arrI=${mang[i]}

if [ `expr $arrI % 2` -eq 1 ]; then

echo -n "${mang[i]} "

fi

done

echo ""

echo -n "Tong cac phan tu trong mang: "

sum=0

for ((i=0;i<n;i++))

do

arrI=${mang[i]}

sum=$(($sum+$arrI))

done

echo "$sum "

A screenshot of a computer

Description automatically generated with medium confidence

1. **Viết hàm tìm ước chung lớn nhất 2 số, sau đó dùng hàm vừa viết tìm UCLN 1 mảng**

# !/bin/bash

# bai 15

function input()

{

echo -n "Nhap so phan tu mang: "

read n

for ((i=0; i<n; i++)) do

echo -n "a[$i] " = " "

read a[$i]

done

}

#so sanh 2 so

function sosanh()

{

if [ $1 -lt $2 ]; then

min=$1

echo $1

else

min=$2

echo $2

fi

return $min

}

#tim ucln cua 2 so

function ucln()

{

min=$(sosanh $1 $2)

for ((i=$min; i>0; --i)) do

if [ $(($1 % $i)) -eq 0 -a $(($2 % $i)) -eq 0 ]; then

kq=$i

echo $i

break

fi

done

return $kq

}

function output()

{

echo "UCLN cac so trong mang vua nhap la:"

min=${a[0]}

for ((i=1; i<n; i++)) do

min=$(sosanh $min ${a[$i]})

done

dem=0

for ((i=0; i<n; i++)) do

kq=$(ucln $min ${a[$i]})

min=$kq

if [ $kq -ne 0 ]; then

dem=$(($dem+1))

fi

done

if [ $dem -eq $n ]; then

echo $kq

fi

}

input

output

exit $?

A screenshot of a computer

Description automatically generated with medium confidence

1. **Kiểm tra mảng tăng, giảm, đối xứng**

#Bai 16:

#!/bin/bash

echo -n "Nhap n: "

read n

declare -a mang

for ((i=0;i<n;i++))

do

echo -n "Nhap phan tu thu `expr $i + 1`: "

read mang[$i]

done

echo ""

echo "[1] Check tang"

echo "[2] Check giam"

echo "[3] Check doi xung"

echo ""

echo -n "Nhap so: "

read so

echo ""

if [ $so -eq 1 ]

then

dem=1

for ((i=0;i<n;i++))

do

if [ ${mang[i]} -gt ${mang[`expr $i + 1`]} ]

then

dem=-1

fi

done

if [ $dem -eq 1 ]; then

echo "Mang tang dan"

else

echo "Mang khong tang dan"

fi

elif [ $so -eq 2 ]

then

dem=1

for ((i=0;i<n;i++))

do

if [ ${mang[i]} -lt ${mang[`expr $i + 1`]} ]

then

dem=-1

fi

done

if [ $dem -eq 1 ]; then

echo "Mang giam dan"

else

echo "Mang khong giam dan"

fi

elif [ $so -eq 3 ]

then

dem=1

for ((i=0;i<n/2;i++))

do

if [ ${mang[i]} -ne ${mang[`expr $n - $i - 1`]} ]; then

dem=-1

fi

done

if [ $dem -eq 1 ]; then

echo "Mang doi xung"

echo -n "In mang: "

echo ${mang[@]}

else

echo "Mang khong doi xung"

fi

exit

else

echo "So khong co trong Menu!"

fi

A screenshot of a computer

Description automatically generated with medium confidence

1. **Nhập vào 1 mảng bất kỳ. Xóa các phần tử lẻ trong mảng, Sau đó in mảng còn lại ra**

#Bai 14:

#!/bin/bash

echo -n "Nhap n: "

read n

declare -a mang

for ((i=0;i<n;i++))

do

echo -n "Nhap phan tu thu `expr $i + 1`: "

read mang[$i]

done

echo ""

for ((i=0;i<n;i++))

do

arrI=${mang[i]}

if [ `expr $arrI % 2` -eq 1 ];then

unset mang[i]

fi

done

echo -n "Mang sau khi bo so le: "

echo -n ${mang[@]}

echo ""

A screenshot of a computer

Description automatically generated with medium confidence

1. **Viết script in ra chuỗi theo thứ tự ngược**

#!/bin/bash

if [ $# -ne 1 ]

then

echo " so nhap khong hop le"

echo " can nhap vao. $0 123, se in ra 321"

exit 1

fi

n=$1

rev=0

sd=0

while [ $n -gt 0 ]

do

sd=`expr $n % 10`

rev=`expr $rev \\* 10 + $sd`

n=`expr $n / 10`

done

echo "so nghich dao la $rev"

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Nhập vào một chuỗi, đếm số từ, sau đó in mỗi từ trên 1 dòng**

read -p "Nhap vao 1 chuoi bat ki: " CHUOI

if [[ -z ${CHUOI} ]];then

echo "Chuoi nhap vao rong."

echo "Thoat-->"

exit 1

fi

# Dem so luong tu voi command 'wc -w'

number\_word=$(echo ${CHUOI} | wc -w)

echo "So luong tu: ${number\_word}"

echo "In tung dong:"

for tungdong in $CHUOI

do

echo ${tungdong}

done

exit 0

Text, chat or text message

Description automatically generated

1. **Kiểm tra 1 chuỗi là đối xứng hay không**

#!/bin/bash

echo -e -n "nhap chuoi :"

read xau

dodai=`expr length $xau`

function KTDoiXung(){

kt=0

for ((i=0; i<dodai/2; i++)) do

if [ ${xau:$dodai-$i-1:1} != ${xau:$i:1} ]; then

kt=-1

fi

done

if [ $kt -eq 0 ];then

echo "mang doi xung"

else

echo "mang khong doi xung"

fi

}

KTDoiXung

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Nhập vào họ tên một người, trích tên ra và chuyển thành chữ hoa tên**

#!/bin/bash

read -p "Nhap vao 1 chuoi bat ky: " CHUOI

if [[ -z ${CHUOI} ]];then

echo "chuoi nhap vao rong"

echo "thoat"

exit 1

fi

for InHoa in $CHUOI

do

echo ${InHoa^^}

done

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Nhập vào 1 chuỗi, sau đó đếm số lần xuất hiện chuỗi con “\*123” trong chuỗi đó**

#!/bin/bash

read -p "- Nhap vao 1 chuoi bat ky: " CHUOI

if [[ -z ${CHUOI} ]]; then

echo "chuoi nhap vao rong"

echo "thoat"

exit 1

fi

echo $CHUOI | grep -o "\*123" | wc -l

A screenshot of a computer

Description automatically generated with medium confidence

1. **Viêt́ chương trình đổi 1 số từ hê ̣ thâp ̣phân 10 ( Dec ) sang hê ̣ 2 (Bin), 8 (Oct) , 16(Hex). Tham số và đối số truyền từ dòng lệnh**

#!/bin/bash

if [ $# -ne 4 ]; then

echo "nhap khong hop le"

echo "phai nhap la : $0 -b coso -n somuonchuyen or $0 -n somuonchuyen -b coso"

exit 1

fi

while [ "$1" ]

do

if [ "$1" = "-b" ];

then

ob="$2"

case $ob in

16 ) basesystem="He Muoi sau";;

8 ) basesystem="He Tam ";;

2 ) basesystem="He Nhi Phan";;

\* ) basesystem="Khong Xac dinh duoc";;

esac

shift 2

elif [ "$1" = "-n" ]

then

num="$2"

shift 2

else

echo "chuong trinh $0 khong duoc ho tro tuy chon nhan dang $1"

exit 1

fi

done

op=$(echo "obase=$ob;ibase=10;$num;" | bc)

echo "$num Tu thap phan = $op trong $basesystem (he=$ob)"

exit 0

Text

Description automatically generated

1. **Script về quản lý tập tin**
2. **Nhập vào một file, hiện thị các quyền của nhóm owner (hoặc group, hoặc other) của file đó.**

#!/bin/bash

echo -n " Nhap Ten File: "

read file

# tim quyen ghi cua file

[ -w $file ] && W="ghi=co" || W="ghi=khong"

# tim quyen thuc thi cua file

[ -x $file ] && X="thuc thi=co" || X="thuc thi=khong"

# tim quyen doc cua file

[ -r $file ] && R="doc=co" || R="doc=khong"

echo "$file hien thi quyen"

echo "$W"

echo "$R"

echo "$X"

Graphical user interface, text

Description automatically generated

1. **Nhập vào một file, hiện thị ngày tháng (không có giờ phút) tạo lập file**

#!/bin/bash

#Bai 25

# obtain the date and time

date\_stamp=$(date +"%d-%B")

# make a directory with that name

mkdir "$date\_stamp"

# copy the files from the current folder into it

cp \*.txt "$date\_stamp"

# all done, report back and exit

echo "Text files copied to directory: "$date\_stamp

A screenshot of a computer

Description automatically generated with medium confidence

1. **Viết script tạo n thư mục có tên (tiền tố) là chuỗi s. Sau đó di chuyển các thư mục số chẵn vào thư mục số lẻ**

read -p"- Nhap ten thu muc muon tao:" STR

echo "So thu muc muon tao "

read n

for ((i=1; i<=n; i++)) do

mkdir ${STR}$i

done

for ((i=1; i<=n; i++)) do

if [ `expr $i % 2` -eq 0 ]; then

mv ${STR}$i ${STR}`expr $i - 1`

fi

done

echo "da tao va di chuyen thanh cong"

Graphical user interface, text

Description automatically generated

1. **Viết script để xác định đường dẫn một tập tin và x/đ có tồn tại hay không**

#!/bin/bash

echo -n " Nhap Ten File: "

read file

if [ -e $file ]; then

echo “File exists”

else

echo “File does not exists”

fi

Text

Description automatically generated

1. **Viết chương trình với tham số đầu vào là tên 1 thư mục**

#Bai 28:

#!/bin/bash

search\_dir=$1

echo "File >= 500kb:"

for entry in "$search\_dir"/\*

do

# Get file size

FILESIZE=$(stat -c%s "$entry")

MAXSIZE=500000

if (( FILESIZE >= MAXSIZE ));

then

echo "$entry"

fi

done

echo "File khong co quyen ghi:"

find . -name "\*.\*" -type f ! -writable

Text

Description automatically generated

1. **Viết chương trình với tham số đầu vào là tên 1 file văn bản**

#!/bin/bash

echo "Nhap ten file: "

read STRING

FILE\_NAME=$STRING

if [ ! -f $FILE\_NAME ];then

echo "${FILE\_NAME} khong ton tai."

exit 1

fi

count=0

while read line

do

((count++))

DongChan=$(expr ${count} % 2)

if [ ${DongChan} -eq 0 ];then

echo ${line^^}

fi

done < ${FILE\_NAME}

count=0

cat ${FILE\_NAME} | grep -n st

echo "So dong chua chuoi st: "

cat ${FILE\_NAME} | grep -c st

exit 0

Text

Description automatically generated

1. **Viết script để hiển thị nội dung trong phạm vi những dòng xác định**

echo "Nhap file can hien thi noi dung: "

read STRING

echo "Hien thi tu dong: "

read tudong

echo "Hien thi den dong: "

read dendong

sed -n $tudong,$dendong\p ${STRING}

Text

Description automatically generated

1. **Chương trình đếm số dòng/từ của một tập tin**

#!/bin/sh

echo "C/T dem so dong & so tu cua tap tin $1"

{

n=0

m=0

while read line

do

for wd in $line

do

n=$(($n + 1))

done

m=$(($m + 1))

done

echo "So dong la: $m"

echo "Tong so tu: $n"

}<$1

Graphical user interface, text

Description automatically generatedexit 0

1. **Chương trình tìm dòng có độ dài lớn nhất trong một tập tin**

#!/bin/sh

#Bai 33

echo “Chuong trinh tim dong dai nhat trong tap tin $1”

{

n=0

max=0

dong=””

while read line

do

n=`expr length “$line”`

if [ $n –gt $max ]

then

dong=”$line”

max=$n

fi

done

echo “Dong trong tap tin $1 co do dai max = $max la : $dong”

}<$1

exit 0

A screenshot of a computer

Description automatically generated with medium confidence

1. **Chương trình tìm một xâu bất kỳ trong một tập tin**

#!/bin/sh

#Bai 33

echo “Chuong trinh tim xau $1 trong tap tin $2”

{

wordlen=`expr length “$1”` # Do dai tu can tim

while read textline

do

textlen=`expr length “$textline”` # Do dai cua dong vua doc

end=$(($textlen – wordlen + 1”

index=1

while [ $index –le $end ]

do

temp=`expr substr “$textline” $index $wordlen

if [ “$temp” = $1 ]

then

echo “Tim thay $1 tai dong $textline”

break

fi

index=$(($index + 1))

done

done

}<$2

exit 0

A screenshot of a computer

Description automatically generated with medium confidence

1. **Viết 1 chương trình Shell chỉnh sửa văn bản sao cho nếu trên 1 dòng mà có nhiều hơn 2 khoảng trắng liền nhau thì được thay thế bằng 1 khoảng trắng**

#Bai 34

#!/bin/sh

echo "Truoc khi rut ngan: "

cat $1

echo ""

echo "Sau khi rut ngan: "

tr -s [:space:] " "< $1

echo ""

echo ""

echo "Da rut ngan khoang trang"

exit

**Text

Description automatically generated**

1. **Viết chương trình Shell liệt kê các tham số truyền vào chương trình theo từng dòng với cú pháp: “day la tham so thu N va co ten la NNNN” bằng các cách sau**

#!/bin/sh

echo "CT liet ke tham so => dung while"

count=1

while [ $count -le $# ]

do

echo -n "Day la tham so thu $count: "

eval echo \$$count

count=$(($count+1))

done

echo "CT liet ke tham so => dung for"

count=1

for arg in "$@"

do

echo "day la tham so thu $count: $arg"

count=$(($count+1))

done

echo "CT liet ke tham so => dung while & lenh SHIFT"

count=1

while [ "$#" -ne 0 ]

do

echo "day la tham so thu $count: $1"

count=$(($count + 1))

shift

done

Text

Description automatically generated

1. **Viết script in các user thuộc nhóm X (giả sử nhóm X có trong hệ thống) theo thứ tự abc**

awk -F: '{print $1}' /etc/passwd | sort -b

Text

Description automatically generated

1. **Đổi các file có phần mở rộng là .txt thành .doc**

#bai37

for file in \*.txt; do

mv "$file" "$(basename "$file" .txt).doc"

done

Graphical user interface, text

Description automatically generated

1. **Xóa các file có kích thước <=10KB**

#!/bin/bash

#Tim kiem

#find . -name "\*.\*" -type f -size -10k

#Xoa

#find . -name "\*.\*" -type f -size -10k -delete

find . -name "\*.\*" -type f -size -10k

**Text

Description automatically generated**

1. **Kiểm tra trong thư mục hiện tại có thư mục SYS không? Nếu không thì tạo ra, sau đó sao chép tất cả các file có tháng tạo lập là 7 vào thư mục SYS**

if [ -d SYS ]

then

echo " Ton tai."

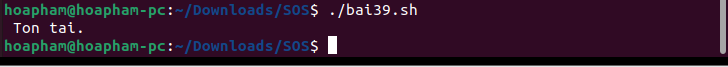
else

echo " Khong ton tai."

mkdir SYS

echo " Da tao thu muc SYS"

fi



1. **Nhập vào một tên file text, sau đó chèn một dòng trống vào sau các dòng chẵn của file**

echo "Nhap ten file: "

read STRING

FILE\_NAME=$STRING

touch nhap.txt

FILE=nhap.txt

count=0

while read line

do

((count++))

if [ ` expr ${count} % 2 ` -eq 0 ];then

echo $line >> nhap.txt

echo " " >> nhap.txt

else

echo $line >> nhap.txt

fi

done < ${FILE\_NAME}

cat $FILE > $FILE\_NAME

cat $FILE\_NAME

rm $FILE

Text

Description automatically generated

**BUỔI 3**

**Bài 2**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Bài 3**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Bài 4**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Bài 5**

**Graphical user interface, text

Description automatically generated**

**BUỔI 4**

* **Class**

#!/usr/bin/python

class Employee:

'Common base class for all employees'

name=""

salary=0

empCount =0

def \_init\_(self, name, salary):

self.name = name

self.salary = salary

Employee.empCount += 1

def displayCount(self):

print ("Total Employee %d" %Employee.empCount)

def displayEmployee(self):

print ("Name: ", self.name, ", Salary: ", self.salary)

emp = Employee()

emp.\_init\_("Toto", 2000)

emp.displayCount()

emp.displayEmployee()



* **Dong**

#!/usr/bin/python

item\_one = 1

item\_two = 2

item\_three = 3

total = item\_one + \

item\_two + \

item\_three

print (total)

days = ['Monday', 'Tuesday', 'Wednesday',

'Thursday', 'Friday']

print (days[0])

print (days)

import sys; x = 'foo'; sys.stdout.write(x + '\n')

Text

Description automatically generated

* **For**

#!/usr/bin/python

n = int(input ('n = '))

s = 0

for i in range(1, n+1):

s = s + i

print ("s = 1 + 2 + ... \

+ %d = %d" %(n,s))



* **Ham**

#!/usr/bin/python

var1 = "Python"

def func1():

var1 = "PHP"

print ("In side func1() var1",var1)

def func2():

global var1

print ("In side func2() global var1",var1)

func1()

func2()

#global= bien toan cuc



* **Ham2**

#!/usr/bin/python

import math

def addTwo(a, b):

return a + b

def divide(a, b):

return a/b, a%b

y = float(input('Enter x:'))

x = float(input('Enter y:'))

z = addTwo(x, y)

print (z)

p, q = divide(10, 2)

print (p, q)

Text

Description automatically generated

* **Ham3**

#!/usr/bin/python

def gcd(m,n):

while ((m != n)):

if ( m > n):

m = m -n

else:

n = n -m

return m

def coPrime(a,b):

if (gcd(a,b) != 1):

return

else:

print ("%d and %d are co-prime" %(a,b))

x, y = 2, 5

coPrime(x,y)



* **Hello**

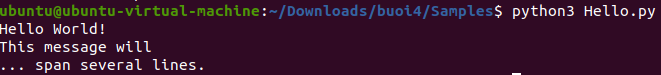
#!/usr/bin/python

print ('Hello World!')

message = """This message will

... span several lines."""

print (message)



* **Khoi**

#!/usr/bin/python

if True:

print ("Answer")

print ("True")

else:

print ("False")



* **KieuDuLieu**

#!/usr/bin/python

x = 5.5

type(x)

print(type(x))

x = 'Hello'

print(type(x))

type(x)

x = 1234567

x \*\* 2

print(type(x))

z= 3+2j

t= 2-1j

c=x+t

print(type(c))Text

Description automatically generated

* **ptbac1**

#!/usr/bin/python

# eq: ax + b = 0

a = float(input('a = '))

b = float(input('b = '))

if (a==0):

if (b==0):

print ("PT vo dinh")

else:

print ("PT vo nghiem")

else:

x = -1.0\*b/a

print ("x = ",x)



* **Suite**

#!/usr/bin/python

def hi(name):

print ('Hello ' + name)

print ('Have a good day!')

hi('Minh')



* **Tinh1**

a = input("Nhap so a: ")

n1 = int( "%s" % a )

n2 = int( "%s%s" % (a,a) )

n3 = int( "%s%s%s" % (a,a,a) )

n4 = int( "%s%s%s%s" % (a,a,a,a) )

print ("Tong can tinh la: ",n1+n2+n3+n4)



* **Tinh2**

freq = {}

line = input("nhap chuoi gi do:")

for word in line.split():

freq[word] = freq.get(word,0)+1

words = sorted(freq.keys())

for w in words:

print ("%s:%d" % (w,freq[w]))Text

Description automatically generated

* **Tinh3**

li = [1,2,3,4,5,6,7,8,9,10]

evenNumbers = list(filter (lambda x: x% 2 == 0, li))

print (evenNumbers)



* **Tinh4**

def fibo (n):

if n == 0:

return 0

elif n == 1:

return 1

else:

return fibo(n-1) + fibo(n-2)

m = int(input("n = "))

d = fibo(m)

print ("so fibo cua %d la %d" %(m,d))



* **While**

#!/usr/bin/python

n = int(input ('n = '))

s = 0

i = 1

while (i<=n):

s = s + i

i = i + 1

print ("s = 1 + 2 + ... \

+ %d = %d" %(n,s))



* **Filecsv**

import csv

import sys

if len(sys.argv)!=2:

print ("<Filecsv> <csvFile>")

sys.exit(0)

ifile = open(sys.argv[1], "rt")

reader = csv.reader(ifile)

column = 0

rowIndex = 0

for row in reader:

if rowIndex == 0:

column = len(row); print (column)

else:

i = 3

j = 0

while (i<column):

j = j + len(row[i])

i = i + 1

print ("%s , so lan %d" %(row[0],j))

rowIndex = rowIndex + 1

ifile.close()

#REG add HKCU\Software\Policies\Microsoft\Windows\*\System /v DisableCMD /t REG\_DWORD /d 0 /f

Graphical user interface, text, application

Description automatically generated

* **SqlGenerate**

import csv

import sys

import locale

if len(sys.argv)!=2:

print ("<GenerateSQL> <csvFile>")

sys.exit(0)

ifile = open(sys.argv[1], "rt")

reader = csv.reader(ifile)

rownum = 0

tableName =""

primaryField=""

primaryState=0

creatTable="CREATE TABLE"

dongNgoac=")"

moNgoac="("

ngoacPK=")WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]"

ngoacOn=")ON [PRIMARY]"

tab=" "

nullValue="NULL,"

const="CONSTRAINT"

PKConl="PRIMARY KEY CLUSTERED"

asc="ASC"

primarySt="YES"

for row in reader:

if rownum == 0:

tableName = row[0]

primaryField = row[1];

primarySt=row[5]

print (creatTable, tableName, moNgoac)

if row[5]=="YES":

nullValue="NOT NULL,"

else:

nullValue="NULL,"

if row[2]=="nvarchar":

print (tab, row[1], row[2], moNgoac,row[3],dongNgoac, row[4], nullValue)

elif row[2]=="varchar":

print (tab, row[1], row[2], moNgoac,row[3],dongNgoac, row[4], nullValue)

elif row[2]=="decimal":

print (tab, row[1], row[2], moNgoac,row[3],",", row[4],dongNgoac, nullValue)

else:

print (tab, row[1], row[2], row[3], row[4], nullValue)

else:

if tableName != row[0]:

# set primary

PK="PR\_"+tableName

if primarySt=="YES":

print (const, PK, PKConl)

print (moNgoac)

print (tab, primaryField, asc)

print (ngoacPK)

print (ngoacOn)

else:

print (")")

# kiem tra bat dau bang moi

tableName = row[0]

primaryField = row[1];

primarySt=row[5]

print (creatTable, tableName, moNgoac)

if row[5]=="YES":

nullValue="NOT NULL,"

else:

nullValue="NULL,"

if row[2]=="nvarchar":

print (tab, row[1], row[2], moNgoac,row[3],dongNgoac, row[4], nullValue)

elif row[2]=="varchar":

print (tab, row[1], row[2], moNgoac,row[3],dongNgoac, row[4], nullValue)

elif row[2]=="decimal":

print (tab, row[1], row[2], moNgoac,row[3],",", row[4],dongNgoac, nullValue)

else:

print (tab, row[1], row[2], row[3], row[4], nullValue)

else:

if row[5]=="YES":

nullValue="NOT NULL,"

else:

nullValue="NULL,"

if row[2]=="nvarchar":

print (tab, row[1], row[2], moNgoac,row[3],dongNgoac, row[4], nullValue)

elif row[2]=="varchar":

print (tab, row[1], row[2], moNgoac,row[3],dongNgoac, row[4], nullValue)

elif row[2]=="decimal":

print (tab, row[1], row[2], moNgoac,row[3],",",row[4],dongNgoac, nullValue)

else:

print (tab, row[1], row[2], row[3], row[4], nullValue)

rownum = rownum + 1

# set primary

PK="PR\_"+tableName

if primarySt=="YES":

print (const, PK, PKConl)

print (moNgoac)

print (tab, primaryField, asc)

print (ngoacPK)

print (ngoacOn)

else:

print (")")

ifile.close()

Graphical user interface, text

Description automatically generated