

SDNB Vaishnav college for women, Chromepet, Chennai-44
I - year B.Sc Computer Science
I-semester Unix Lab programs (UCS/CP/1002)

Exercise 1: write a shell script, which gets executed the moment the user logs in, it should display the message GOOD MORNING / GOOD AFTERNOON / GOOD EVENING depending on the time and user logs in.

program

```
# based on the login time say GM/ GAN/GE
echo "your name:"
read n
current_time=$(date +%H)
current_session=$(date +%p)
echo "currently the time is $current_time o'clock $current_session"
if [ $current_time -lt 12 ];
then
    echo " GOOD MORNING $n"
elif [[ $current_time -gt 12  && $current_time -lt 16 ]];
then
    echo "GOOD AFTERNOON $n"
else
    echo "GOOD EVENING $n"
fi
exit
```

output:

```
administrator@ubuntu:~/rr$ ./logtime.sh
your name:
aswini
currently the time is 08 o'clock AM
GOOD MORNING aswini
```

Exercise 2: Write a shell script to search a file from the current directory in any of the sub-directories and report the path

program

```
# program to search a file
echo "Enter the name of the file"
read file1
find ../ -name $file1 -print
```

output

```
administrator@ubuntu:~/rr$ ./ex2.sh
Enter the name of the file
palin.sh
../rr/palin.sh
```

Exercise 3: Write a function GO which would change the \$prompt to the current directory name in which you are working. Thus if you are working in \$usr\$acc the prompt should be \$usr\$acc

program

```
# Change the prompt
fun_go()
{
    echo "this function changes the prompt to present working directory"
    echo
    PS1='[$u@$h]$w$'
    echo $PS1
}
fun_go
```

output

```
administrator@ubuntu:~/rr$ ./ex33.sh
this function changes the prompt to present working directory

administrator@ubuntu:~/rr$
```

exercise 4: write a shell script to find whether an input year is leap year or not

program

```
#program to find the year is leap or not
echo "enter the year"
read y
ans=`expr $y % 4`
if [ $ans -eq 0 ]
then
echo "$y is a leap year"
else
echo "$y is not a leap year"
fi
```

output

```
administrator@ubuntu:~/rr$ ./leapyear.sh
enter the year
2012
2012 is a leap year
administrator@ubuntu:~/rr$ ./leapyear.sh
enter the year
2001
2001 is not a leap year
```

Exercise 5: Write a shell script to find out the greatest among three inputs.

Program

```
#program to find the greatest among three digits
echo "enter the first number"
read no1
echo "enter the second number"
read no2
echo "enter the third number"
read no3
if [ $no1 -gt $no2 -a $no1 -gt $no3 ]
then
echo "$no1 is greatest"
elif [ $no2 -gt $no3 ]
then
echo "$no2 is greatest"
else
echo "$no3 is greatest"
fi
```

output

```
administrator@ubuntu:~/rr$ ./greatest.sh
enter the first number
45
enter the second number
23
enter the third number
67
67 is greatest
```

Exercise 6: write a shell script to check if the input number is prime or not

program

```
#program for prime number checking
echo "enter the number"
read num
i=2
while [ $i -lt $num ]
do
k=`expr $num % $i`
if [ $k -eq 0 ]
then
echo "the number is not a prime number"
exit
fi
i=`expr $i + 1`
done
echo "the number is prime"
```

output

```
administrator@ubuntu:~/rr$ ./prime.sh
enter the number
45
the number is not a prime number
administrator@ubuntu:~/rr$ ./prime.sh
enter the number
5
the number is prime
```

Exercise 7: Write a shell script to check if the input string is palindrome or not

program

```
# program to check for palindrome
echo "enter the string for palindrome check"
read str1
len=`echo $str1 | wc -c`
len=`expr $len - 1`
echo "the string $str1 length is = $len"
while [ $len -gt 0 ]
do
rev=$rev`echo $str1 |cut -c $len`
len=`expr $len - 1`
done
echo "the reversed string is $rev"
if [ $str1 != $rev ]
then
echo "the given string is not a palindrome"
else
echo "the given string is a palindrome"
fi
```

output

```
administrator@ubuntu:~/rr$ ./palin.sh
enter the string for palindrome check
madam
the string madam length is = 5
the reversed string is madam
the given string is a palindrome
administrator@ubuntu:~/rr$ ./palin.sh
enter the string for palindrome check
computer
the string computer length is = 8
the reversed string is retupmoc
the given string is not a palindrome
```

Exercise 8: Program to reverse the given number

Program

```
# reverse the number
```

```
echo "enter the number for reversal"
```

```
read str1
```

```
len=`echo $str1 | wc -c`
```

```
len=`expr $len - 1`
```

```
echo "the string $str1 length is = $len"
```

```
while [ $len -gt 0 ]
```

```
do
```

```
rev=$rev`echo $str1 |cut -c $len`
```

```
echo $rev
```

```
len=`expr $len - 1`
```

```
done
```

```
echo "$rev"
```

output

```
administrator@ubuntu:~/rr$ ./revstring.sh
```

```
enter the number for reversal
```

```
12345
```

```
the string 12345 length is = 5
```


5
54
543
5432
54321
54321

Exercise 9: Write a shell script to print all the multiplication tables(up to 10) between 2 given number

program

```
#program for multiplication tables(up to 10) between 2 given number
echo "enter the table1"
read n
echo "enter the table2"
read n1

while [ $n -le $n1 ]
do
count=1
echo "printing $n table"
while [ $count -le 10 ]
do
table1=`expr $count \* $n`
echo "$count * $n =" $table1
count=`expr $count + 1`
done
n=`expr $n + 1`
done
```

output

```
administrator@ubuntu:~/rr$ ./multtable.sh
enter the table1
2
enter the table2
4
printing 2 table
1 * 2 = 2
2 * 2 = 4
3 * 2 = 6
```

$$4 * 2 = 8$$

$$5 * 2 = 10$$

$$6 * 2 = 12$$

$$7 * 2 = 14$$

$$8 * 2 = 16$$

$$9 * 2 = 18$$

$$10 * 2 = 20$$

printing 3 table

$$1 * 3 = 3$$

$$2 * 3 = 6$$

$$3 * 3 = 9$$

$$4 * 3 = 12$$

$$5 * 3 = 15$$

$$6 * 3 = 18$$

$$7 * 3 = 21$$

$$8 * 3 = 24$$

$$9 * 3 = 27$$

$$10 * 3 = 30$$

printing 4 table

$$1 * 4 = 4$$

$$2 * 4 = 8$$

$$3 * 4 = 12$$

$$4 * 4 = 16$$

$$5 * 4 = 20$$

$$6 * 4 = 24$$

$$7 * 4 = 28$$

$$8 * 4 = 32$$

$$9 * 4 = 36$$

$$10 * 4 = 40$$

Exercise 10: Write a shell script to generate the series 1, 5, 2, 10, 3, 15...50
program

```
# program to find the sum of 1, 5, 2, 10, 3, 15, ... 50
echo "enter the number"
read n
first=1
second=5
while [ $second -le $n ]
do
printf "$first, $second, "
first=`expr $first + 1`
second=`expr $second + 5`
done
```

output

```
administrator@ubuntu:~/rr$ ./sumseries.sh
enter the number
50
1, 5, 2, 10, 3, 15, 4, 20, 5, 25, 6, 30, 7, 35, 8, 40, 9, 45, 10, 50
```

Exercise 11: Write a shell script to find out the sum of series $1/2+2/3+..+(n-1)/n$

program

```
# program to find the sum of 1/2+2/3+.. +(n-1)/n
echo "enter the number"
read n
sum=0.0
num=1.0
deno=2.0
count=2
y=0.0
while [ $count -le $n ]
do
y=$(echo "$num / $deno" | bc -l)
printf "%10.5f\n" $y
sum=`echo $sum + $y | bc`
deno=`echo $deno + 1.0 | bc`
num=`echo $num + 1.0 | bc`
count=`echo $count + 1 | bc`
done
echo "numerator $num denominator $deno"
printf "%10.5f " $sum
```

output

```
administrator@ubuntu:~/rr$ ./sumseries1.sh
enter the number
7
0.50000
0.66667
0.75000
0.80000
0.83333
0.85714
answer = 4.40714
```

Exercise 12: Write a shell script to calculate the net salary of an employee in a particular month considering various allowances (TA, DA, HRA) and deductions (Income tax, Providend fund) as TA=15% of basic, DA=2% of basic , HRA=10% of basic, Income tax=5% of basic and providend fund=10% of basic

```
program
# program for payslip
echo "enter the name of the employee"
read name
month=$(date +"%h %Y")
echo $name, $month
echo "enter the basic pay"

read basic
da=`echo $basic ¥* 0.02 | bc -l`
hra=`echo $basic ¥* 0.10 | bc -l`
ta=`echo $basic ¥* 0.15 | bc -l`
it=`echo $basic ¥* 0.05 | bc -l`
pf=`echo $basic ¥* 0.10 | bc -l`
allow=`echo $da + $hra + $ta | bc -l`
ded=`echo $it + $pf | bc -l`
net=`echo $basic + $allow | bc -l`
gross=`echo $net - $ded | bc -l`
echo " payslip for the employee : $name"
echo " basic pay is : $basic"
echo " Allowances are : "
echo " DA is : $da"
echo " TA is : $ta"
echo " HRA is : $hra"
echo " Deductions are : "
echo " IT is : $it"
echo " PF is : $pf"
echo " Net Salary is : $net"
echo " Gross Salary is : $gross"
```

output

```
administrator@ubuntu:~/rr$ ./payslip.sh
```

enter the name of the employee

sumi

sumi, Oct 2013

enter the basic pay

8900

payslip for the employee : sumi

basic pay is : 8900

Allowances are :

DA is : 178.00

TA is : 1335.00

HRA is : 890.00

Deductions are :

IT is : 445.00

PF is : 890.00

Net Salary is : 11303.00

Gross Salary is : 9968.00

Exercise 13: Write a shell script which receives two files names as arguments. Check whether the file contents are same or not. If same delete the second one.

program

```
# program to compare two files and check if the content are same.
# if same delete the second one.
first=$1
second=$2
x=`diff $first $second`
if [ $? = 0 ]
then
echo "both files are equal"
rm $second
else
echo "there is difference"
echo "the difference is $x"
fi
if [ -e $second ]
then
echo "file is not deleted"
else
echo "file is deleted"
fi
```

output :

content of test

welcome to csc

content of test4

welcome to csc

content of test2

welcome

csc

administrator@ubuntu:~/rr\$./filesame.sh test test2

there is difference

the difference is 1c1,2

< welcome to csc

```
> welcome
> csc
file is not deleted
administrator@ubuntu:~/rr$ ./filesame.sh test test4
both files are equal
file is deleted
```

Exercise 14: program to accept two files and check whether both exists.
if the second file exists then content of first file should be appended to it
if the second file does not exists then create a new file with the contents of
first file.

Program

```
# Exercise 14: program to accept two files and check if they exists.
# if the second file exists then content of fist file should be appended to it
# if the second file does not exists then creat a new file with the contents # of
first file.
first=$1
second=$2
if [ -e $first ]
then
echo "first file $first exists"
fi
if [ -e $second ]
then
echo "second file $second exists"
echo "first file is appended to the content of second"
cat $first >> $second
else
cat $first > $second
fi
```

output

```
administrator@ubuntu:~/rr$ ./fileapp.sh test test5
first file test exists
```

```
administrator@ubuntu:~/rr$ ./fileapp.sh test test5
first file test exists
second file test5 exists
first file is appended to the content of second
```


Exercise 15: a) List of all files in the current directory to which you have read, write, and execute permission.

Program

```
# exercise 15: a ) Evaluate the status of a file
x=".*"
for y in $x
do
if [ -f $y ]
then
if [ -s $y ]
then
echo "$y is a file and size is >0"
fi
fi
if [ -d $y ]
then
echo "$y is a directory"
fi
if [ -r $y ]
then
if [ -w $y ]
then
if [ -x $y ]
then
echo "$y has read, write and execute permission"
else
echo "$y has read and write permission"
fi
else
echo " $y has read permission only"
fi
else
echo "$y has no permission"
fi
done
```

Output

./test has read and write permission
./test2 is a file and size is >0
./test2 has read and write permission
./test3 is a file and size is >0
./test3 has read and write permission
./test4 is a file and size is >0
./test4 has read and write permission
./test5 is a file and size is >0
./test5 has read and write permission
./unixoutput.odt is a file and size is >0
./unixoutput.odt has read and write permission
./untileg1.sh is a file and size is >0
./untileg1.sh has read, write and execute permission
./whileeg1.sh is a file and size is >0
./whileeg1.sh has read, write and execute permission
./x is a directory
./x has read, write and execute permission

Exercise 15: b) Receive any number of filenames as arguments and check whether the argument supplied is a file or directory. If it is a directory it should be appropriately reported. If it is a file then name of the file as well as the number of lines present in it should be reported.

Program

```
# exercise 15: a ) Evaluate the status of a file
x="./*"
for y in $x
do
if [ -f $y ]
then
if [ -s $y ]
then
echo "$y is a file and size is >0"
fi
fi
if [ -d $y ]
then
echo "$y is a directory"
fi
len=`echo $y | wc -lc`
echo "$y contains $len lines"
done
```

output

```
./caseeg1 is a file and size is >0
./caseeg1 contains      1      10 lines
./caseeg1.sh is a file and size is >0
./caseeg1.sh contains      1      13 lines
./caseeg2.sh is a file and size is >0
./caseeg2.sh contains      1      13 lines
./caseeg3.sh is a file and size is >0
./caseeg3.sh contains      1      13 lines
```

```
./copyprg.sh is a file and size is >0
./copyprg.sh contains      1      13 lines
./dateformat.sh is a file and size is >0
./dateformat.sh contains   1      16 lines
./ex2.sh is a file and size is >0
./ex2.sh contains          1      9 lines
```

Exercise 16: Create file called Test, which contains sample data as follows:

```
A00001 Shanthi 80
A00007 Arun 70
S00005 Karthi 50
A00004 sunitha 90
A00002 kanimozhi 45
```

Answer the following questions based on the above:

- Display the contents of the file sorted according to the marks in the descending order.
- Display the names of the students in the alphabetical order ignoring the cases.
- Display the list of students who have scored marks between 60 and 80
- Display the list of students and their register number.

program

```
# exercise 16
cat test
echo "displaying the contents in decending order of marks"
sort -r -k 3 test
echo "displaying the contents in ascending order of names"
sort -d -k 2 test
echo "-----"
echo "displaying the students list who have scored between 60 & 80"
echo "-----"
while read roll name mark
do
if [[ $mark -ge 70 && $mark -le 80 ]]
then
echo $roll $name $mark
fi
done < test
echo "-----"
format="%10s%10s%3s¥n"
echo "ALL STUDENTS AND THEIR REGISTER NUMBERS"
while read roll name mark
do
```

```
printf "$format" $name $roll
done < test
echo "-----"
exit
```

output

```
outadministrator@ubuntu:~/rr$ ./exer16.sh
A00001 Shanthi 80
A00007 Arun 70
S00005 Karthi 50
A00004 sunitha 90
A00002 kanimozhi 45
displaying the contents in decending order of marks
A00004 sunitha 90
A00001 Shanthi 80
A00007 Arun 70
S00005 Karthi 50
A00002 kanimozhi 45
displaying the contents in ascending order of names
A00007 Arun 70
A00002 kanimozhi 45
S00005 Karthi 50
A00001 Shanthi 80
A00004 sunitha 90
-----
displaying the students list who have scored between 60 & 80
-----
A00001 Shanthi 80
A00007 Arun 70
-----
ALL STUDENTS AND THEIR REGISTER NUMBERS
  Shanthi    A00001
    Arun    A00007
    Karthi   S00005
  sunitha   A00004
kanimozhi   A00002
-----
```