Write a script to reverse a number given by user

# Reverse the number

echo enter n

read n

**num**=0

**while** [ **$n** –**gt** 0 ]

**do**

**num**=$(expr **$num** \\* 10)

**k**=$(expr **$n** % 10)

**num**=$(expr **$num** + **$k**)

**n**=$(expr **$n** / 10)

**done**

echo number is **$num**

Write a script to find the smallest of three numbers as well as largest among three

**echo "enter a: "  
read a  
echo "enter b : "  
read b  
echo "enter c : "  
read c  
s=$a  
if [ $b -lt $s ]  
then  
s=$b  
fi  
if [ $c -lt $s ]  
then  
s=$c  
fi  
echo Smallest of $a $b $c is $s**

#shell script to find the greatest of three numbers

Echo "Enter Num1"

read num1

echo "Enter Num2"

read num2

echo "Enter Num3"

read num3

**if** [ $num1 –gt $num2 ] && [ $num1 –gt $num3 ]

**then**

echo$num1

**elif** [ $num2 –gt $num1 ]&& [ $num2 –gt $num3 ]

**then**

echo $num2

**else**

echo $num3

**fi**

# Sorting

#!/bin/bash

declare nos[5]=(4 -1 2 66 10)

# Prints the number befor sorting

echo "Original Numbers in array:"

for (( i = 0; i<= 4; i++ ))

do

echo ${nos[$i]}

done

#

# Now do the Sorting of numbers

#

for (( i = 0; i<= 4 ; i++ ))

do

for (( j = $i; j <= 4; j++ ))

do

if [ ${nos[$i]} -gt ${nos[$j]} ]; then

t=${nos[$i]}

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

nos[$j]=$t

fi

done

done

#

# Print the sorted number

#

echo -e "\nSorted Numbers in Ascending Order:"

for (( i=0; i<= 4; i++ ))

do

echo ${nos[$i]}

done

#!/bin/bash

IFS=' ' read -raarr -p "Enter numbers: "

Enter numbers: 4 -1 2 66 10

sort -n <(printf "%s\n" "${arr[@]}")

-1

2

4

10

66

* IFS=' ' to make read all number delimited by space
* 'read -ra` to read all numbers in an array
* sort -n to sort numbers numerically
* printf "%s\n" "${arr[@]}" to print each element of array in separate line
* <(printf "%s\n" "${arr[@]}") is process substitution that make it printf command behave like a file for sort -n command.

Write script that prints names of all sub directories present in the current directory.

**Approach:**

1. Go to the root directory or any target directory where you want to get its sub-directories.
2. Type in the following program.
3. We are using ‘echo’ so that we get a simplified output.
4. ‘ls’ command in Linux lists all the files and directories.
5. ‘-d’ option in ls command specifies that we need to list only the directories.
6. We are using the expression ‘\*/’ so that we get only the directories under the current directory level and not going any level in depth further.

**Program:**

# Shell Script to list all sub-directories

# present in a current folder

# echo prints a message for user on screen

echo "List of sub-directories present in this Folder - "

# Following command lists all sub directories

# '\*/' will only match directories under the current directory

ls -d \*/

**C Program to Simulate the ls Command in Linux Operating system**

This is a c program to do Simulation of ls Command in linux. The ls command lists all the contents of the directory including filse and sub-directories. The following c program in simulates the ls command.

#include<stdio.h>

#include<dirent.h>

main()

{

char dirname[10];

DIR\*p;

struct dirent \*d;

printf("Enter directory name\n");

scanf("%s",dirname);

p=opendir(dirname);

if(p==NULL)

  {

  perror("Cannot find directory");

  exit(-1);

  }

while(d=readdir(p))

  printf("%s\n",d->d\_name);

}