Lab 2

Set execute permission: chmod +x prog.sh

Execute the file prog.sh: sh prog.sh

is comment character

sh increment:– i=$((i+1)) or ((i++))

#!/bin/bash ((i++)) will work in bash and sh

bash prog.sh

Control Statement:

If-else

Program 1:

```
echo "Enter a number:"
read a
if((a>=0))
then
        echo "$a is +ve"
else
        echo "$a is negative"
fi
```

compare options (man test)

-f, -e, -d, -gt, -le, -ne

x=`expr $1 + $2`

x=`expr any_expression|bc –l` #-l for floating point computations

Floating point arithmetic:

Program2

```
a=10.5;b=3.5
c=`echo $a + $b | bc -l`
d=`echo $a - $b |bc -l`
e=`echo $a \* $b|bc -l`
f=`echo $a / $b|bc -l`
echo "$a+$b=$c"
echo "$a-$b=$d"
echo "$a*$b=$e"
echo "$a/$b=$f"
```

for loop:

Program3

```
for((i=0;i<5;i++))
do
        echo $i
done
for i in “abc” “def” “xyz”
do
        echo $i
done
IFS=”/”   #Internal Field Seperator
read x                          #abc/def/ghi/klm   :value for variable x
for i in $x
do
        echo $i
done
```

while and until loop:

Program 4:

```
$i=0
echo “while loop: executes if condition is true”
while((i<5))
do
echo $i
        ((i++))
done
$i=0
echo “until loop: executes if condition is false”
until((i>5))
do
        echo “$i”
        ((i++))
done
```

i=$((x+y)) // performs only int operations

((i++))// performs only int operations

String Comparison

Program 5:

```
if [ “$str1” = “$str2” ];then
        echo “Strings are equal”
fi
if [ “$str1” != “$str2” ];then
        echo “Strings are not equal”
fi
if [ -n  “$str1” ];then
        echo “Length of str1 is greater then 0”
fi
if [ -z “$str1” ];then
        echo “Length of str1 is equal to 0”
fi
```

+ - Match one or more times, ? – Match zero or one, * - match zero or more

Program 6:

```
if [ “$word” = [yY]* ];then
        echo “word starts with y or Y”
fi
if [ “$word” = [aeiouAEIOU]* ];then
        echo “Word starts with vowel”
fi
if  [ “$word” = [0-9]* ];then
        echo “Word starts with a digit”
fi
if [ “$word” = [a-zA-Z][a-zA-Z][a-zA-Z]* ];then
        echo “First three characters are alphabets”
fi
if [ “$word” = [!a-z]* ] or if [ “$word” = [^a-z]* ]
then
        echo “Word does not start with lower case letter”
fi
```

Switch-case:

```
case word in
  pattern1)
     Statement(s) to be executed if pattern1 matches
     ;;
  pattern2)
     Statement(s) to be executed if pattern2 matches
     ;;
  pattern3)
     Statement(s) to be executed if pattern3 matches
     ;;
esac
```

```
#!/bin/sh
echo "Enter a number between 1 and 10. "
read NUM
case $NUM in
        1) echo "one" ;;
        2) echo "two" ;;
        3) echo "three" ;;
        4) echo "four" ;;
        5) echo "five" ;;
        6) echo "six" ;;
        7) echo "seven" ;;
        8) echo "eight" ;;
        9) echo "nine" ;;
        10) echo "ten" ;;
        *) echo "INVALID NUMBER!" ;;
esac
```

```
FRUIT="kiwi"
case "$FRUIT" in
   "apple") echo "Apple pie is quite tasty."
   ;;
   "banana") echo "I like banana nut bread."
   ;;
   "kiwi") echo "New Zealand is famous for kiwi."
   ;;
esac
```

command line arguments

echo $0- returns the first command line argument: always the filename

echo $# - returns number of command line arguments

echo $@- stores all command line arguments

echo ‘$a*0.1’|bc –l

bc –l<<$a*0.1

IFS="/" command line argument is a/b/c/d/e/f/g/h
IFS: Internal Field Separator

```
for i in $@
do
        echo $i
done
```

```
$test.sh a b c d e f g h
echo "$1,$2,$3,$4,$5,$6,$7,$8,$9"
```

```
shift 2   # shifts command line args by 2 places
```