Linux assignment –9

Name: Pranav H Raikar sec: Cy(3B)

USN: ENG24CY0041 Roll no: 39

1. **Write a shell script using if … elseto check if a number is even or odd.**

Ans:

```
#!/bin/bash
echo "enter a number"
read num
if [ $((num % 2)) -eq 0 ]
then
echo "$num is even."
else
echo "$num is odd."
fi
```

Here the script will cheeks thet if the reminder when divivding the number by2 is zero if the condition setisfyes then it is even other wise it will be odd.

2. **Explain the difference between if and case statements in bash.**

Ans: the both if and case statements are used to decision making statements in the programming language.

If : the if ststement is used when we have to check the numarical or logical cnditions or when expression involve opertors.

Ex: if [$num -gt 10];

then

echo "greater than 10"

fi

Case : the case statment is used to cheek better to use multiple possible value of single variable,it also ac like switch statments.

```
case $choice in
    1) echo "option 1";;
    2) echo "option 2";;
   *) echo "invalid choice";;
esac
```

3. **Write a script to find thelargest of three numbers entered by the user.**

Ans:

```
#!/bin/bash
echo "enter three numbers"
read a b c
if [ $a -ge $b ] && [ $a -ge $c ];
 then
echo "$a is the largest number"
elif [ $b -ge $a ] && [ $b -ge $c ];
 then
echo "$b is the largest number"
else
 echo "$c is the largest number"
 Fi
```

The above script will cheek which is the largest nuber of a given number by compare themself by if else statements.and then pint the output.

4. **How do you use afor loop to treverse an array in bash? Give an example.**

Ans:

```
#!/bin/bash
arr=(123 "abs" -2.3 'A' 23.56 0)
echo "array elements are:"
for item in "${arr[@]}"
do
echo $item
Done
```

Here the for loop goes through evey element of the given array and print the one by one element in the error to the output display. This is the simplest way to travers and display all element in array by the bash script programming language.

5. **Write a shell script to loop through all filein the current directory and display their names.**

Ans;

```
#!/bin/bash
echo "files   the current directory"
for file in *
do
echo $file
Done
```

Here the * symbol represt the all files in the current directory the for loop will goes through every element and print s the element in the array.

6. **What is the difference between while and until loops in bash?**

Ans: the both while and untill loops are used to repeat a set of connands but their condition worsks opposite to each other

While loop: the while loop will keeps running untill the condition get satisfied or condition gets truee other wise the loop statement will terminate.

Example: count=1

```
while [ $count -le 5 ]
 do
echo $count
((count++))
 done
```

Until: it will ruunig untill the condition become true that is it runs while the condition is false.

Example: count=1

```
 until [ $count -gt 5 ]
 do
echo $count
((count++))
 done
```

So the main difference is the while runs when ever condition gets true and until loop will terminte when condition get false.

7. **Write a countdown timer script using a while loop**

Ans:

```
#!/bin/bash
 echo "enter countdown time seconds:"
 read time
 while [ $time -gt 0 ]
 do
echo "Time left: $time seconds"
 sleep 1
((time--))
 done
echo "Time is over"
```

Here the script takes a number of secounds from the user and coundown by one by one every secound using the sleep command to delay for 1 seconds prints.

8. **How do you use break and continue statements in loops? Give examples.**

Ans: the brake and the contine statement are used to execute output accourding to user .

Break : thre break statement is used to exit the loop complitly when a certiain condition get satisfied or completed succesfully the break statement will stop the loop condition and getout from the user.

Example: for i in {1..10}

```
do
if [ $i -eq 5 ];
then
break
fi
echo $i
done
```

Continue: the continue statement will skip the current iterration and then moves to the next iteration and again it will not skip other iterration only the given value with the contineue statement will skipped and other will run normallly.

Example: for i in {1..5}

```
do
if [ $i -eq 3 ];
then
continue
fi
echo $i
Done
```

9. **Write a script to check if a file exists or not using the if and else loop.**

Ans:

```
#!/bin/bash
echo "enter file name:"
read file
```

```
if [ -e "$file" ]
then
echo "file '$file' exists."
else
echo "file '$file' does not exist."
 fi
```

The –e option cheeks wehether the file is exist or not in the directory. If it exist or found it will print the successfull of finding message ,otherwise it will show that the file doesn't exit.

10. Write a script to calculate factorial of a number using for loop.

Ans:

```
#!/bin/bash
echo "enter a number "
read num
fact=1
for ((i=1; i<=num; i++))
do
fact=$((fact * i))
done
echo "eactorial of $num is $fact"
```

The script written in the above seection is to multiplies all number from 1 to the given number using a for loop if the input number is 3 then it will multiple 3*2*1 and return 6