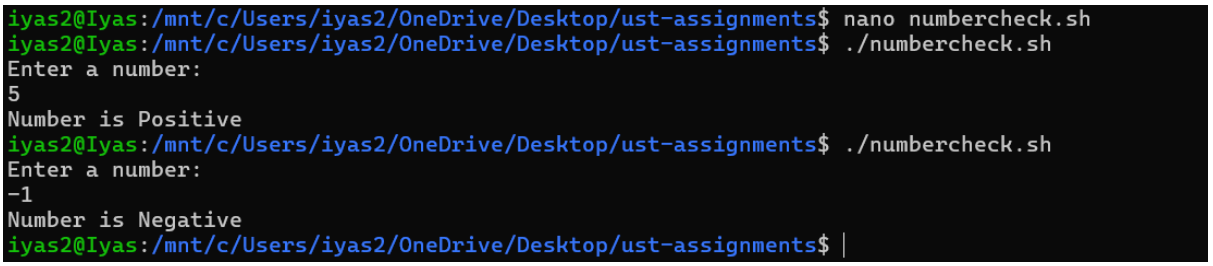
IYAS K
iyas2458@gmail.com

1️⃣ Number Classification (if + function)

Write a shell script that uses a function to check whether a number is positive, negative, or zero using if.

```
#!/bin/sh

check_number() {
   if [ $1 -gt 0 ]; then
      echo "Number is Positive"
   elif [ $1 -lt 0 ]; then
      echo "Number is Negative"
   else
      echo "Number is Zero"
   fi
}

echo "Enter a number:"
read num

check_number $num
```



2️⃣ Factorial of a Number (while + function)

Write a shell script to calculate the factorial of a number using a while loop inside a function.

```
#!/bin/sh

factorial() {
   n=$1
   fact=1
```

```
    while [ $n -gt 1 ]
    do
        fact=$((fact * n))
        n=$((n - 1))
    done

    echo "Factorial is: $fact"
}

echo "Enter a number:"
read num

factorial $num
```

```
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$ nano factorial.sh
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$ ./factorial.sh
Enter a number:
5
Factorial is: 120
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$
```

3️⃣ Sum of N Numbers (for + function)

Write a shell script to calculate the sum of numbers from 1 to N using a for loop inside a function.

```
#!/bin/sh

sum_n() {
    n=$1
    sum=0

    for i in $(seq 1 $n)
    do
        sum=$((sum + i))
    done

    echo "Sum is: $sum"
}

echo "Enter a number:"
read num

sum_n $num
```

```
Enter a number:
6
Sum is: 21
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$ ./sumofn.sh
Enter a number:
12
Sum is: 78
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$
```

4️⃣ Prime Number Check (if + for + function)

Write a shell script that uses a function to check whether a given number is prime using if and for.

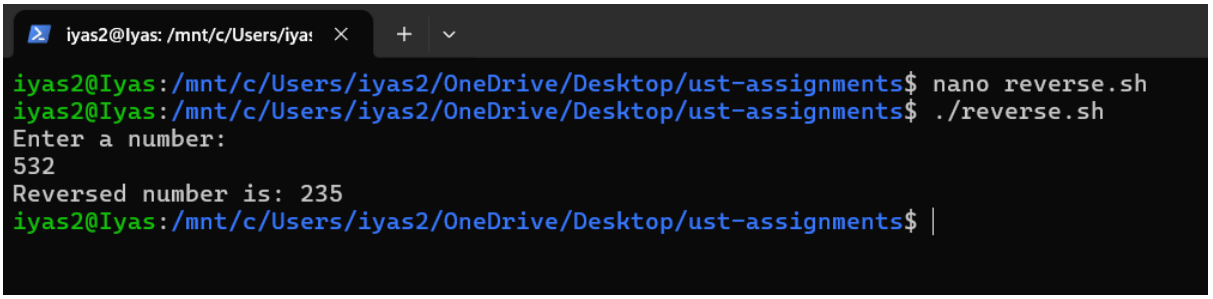
```
#!/bin/sh

check_prime() {
  num=$1
  flag=0

  if [ $num -le 1 ]; then
    echo "Not a Prime Number"
    return
  fi

  for i in $(seq 2 $((num - 1)))
  do
    if [ $((num % i)) -eq 0 ]; then
      flag=1
      break
    fi
  done

  if [ $flag -eq 0 ]; then
    echo "Prime Number"
  else
    echo "Not a Prime Number"
  fi
}

echo "Enter a number:"
read n

check_prime $n
```

```
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$ ./primecheck.sh
Enter a number:
5
Prime Number
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$ ./primecheck.sh
Enter a number:
6
Not a Prime Number
iyas2@Iyas:/mnt/c/Users/iyas2/OneDrive/Desktop/ust-assignments$
```

5️⃣ Reverse a Number (while + if)

Write a shell script to reverse a number using a while loop and if conditions where required.

```
#!/bin/sh

echo "Enter a number:"
read num

rev=0

while [ $num -gt 0 ]
do
   rem=$((num % 10))
   rev=$((rev * 10 + rem))
   num=$((num / 10))
done

if [ $rev -ge 0 ]; then
   echo "Reversed number is: $rev"
fi
```

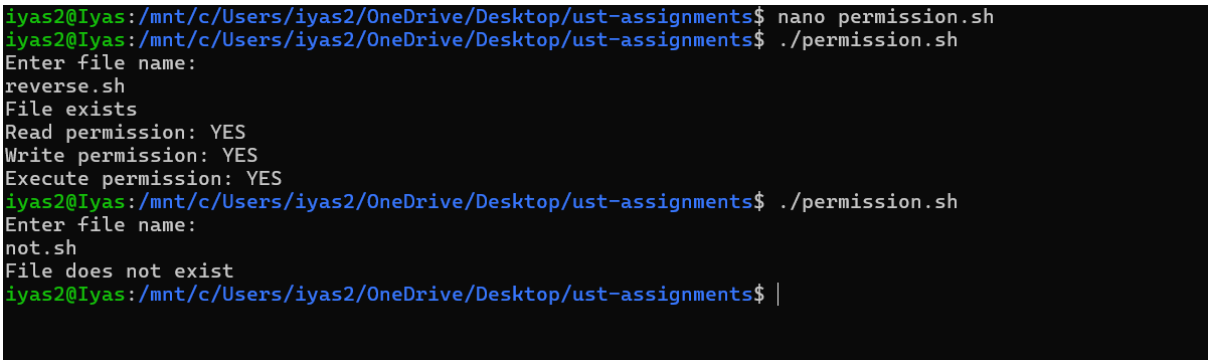


6️⃣ File Existence & Permission Check (if + function)

Write a shell script that uses a function to check if a file exists and whether it has read, write, and execute permissions.

```
#!/bin/sh

check_file() {
```

```
    file=$1

    if [ -e "$file" ]; then
        echo "File exists"

        if [ -r "$file" ]; then
            echo "Read permission: YES"
        else
            echo "Read permission: NO"
        fi

        if [ -w "$file" ]; then
            echo "Write permission: YES"
        else
            echo "Write permission: NO"
        fi

        if [ -x "$file" ]; then
            echo "Execute permission: YES"
        else
            echo "Execute permission: NO"
        fi
    else
        echo "File does not exist"
    fi
}

echo "Enter file name:"
read fname

check_file "$fname"
```



7️⃣ Menu-Driven Calculator (while + case + function)

Write a shell script that shows a menu in a while loop and performs add, subtract, multiply, divide using functions.

```
#!/bin/sh

add() {
   echo "Result: $((a + b))"
}

sub() {
   echo "Result: $((a - b))"
}

mul() {
   echo "Result: $((a * b))"
}

div() {
   if [ $b -ne 0 ]; then
      echo "Result: $((a / b))"
   else
      echo "Division by zero not allowed"
   fi
}

while true
do
   echo "1. Add"
   echo "2. Subtract"
   echo "3. Multiply"
   echo "4. Divide"
   echo "5. Exit"
   echo "Enter choice:"
   read ch

   if [ $ch -eq 5 ]; then
      break
   fi

   echo "Enter first number:"
   read a
   echo "Enter second number:"
   read b

   case $ch in
      1) add ;;
      2) sub ;;
```

```
        3) mul ;;
        4) div ;;
        *) echo "Invalid choice" ;;
    esac
done
```