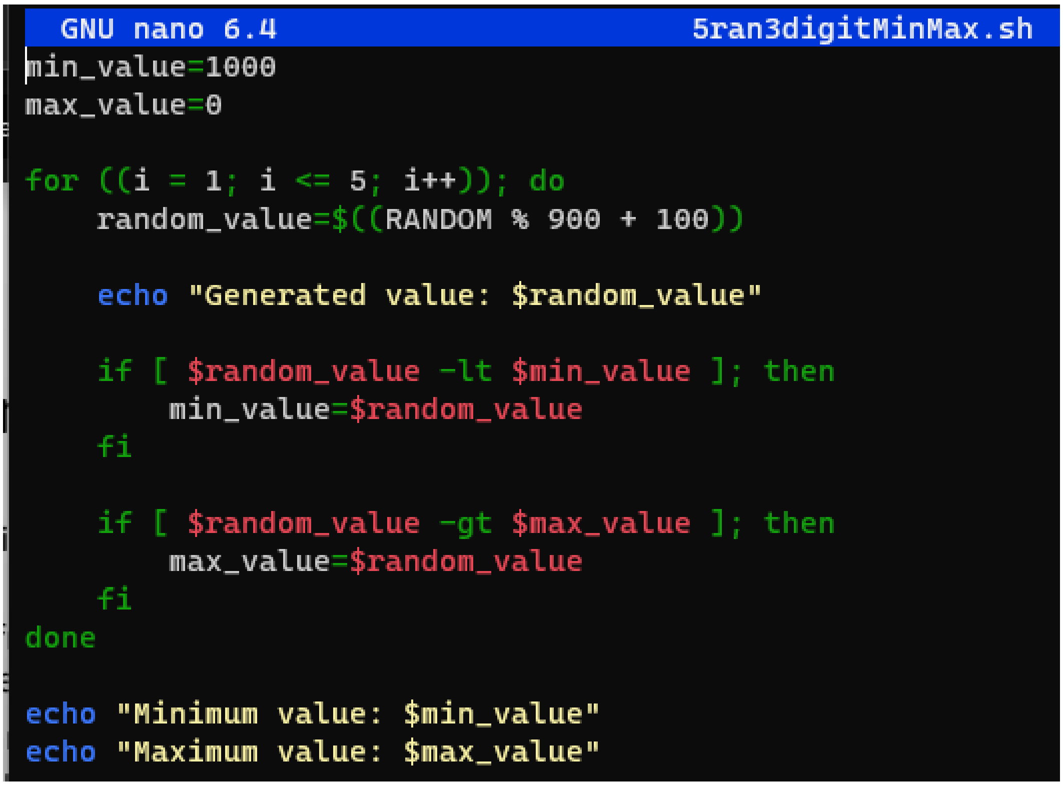
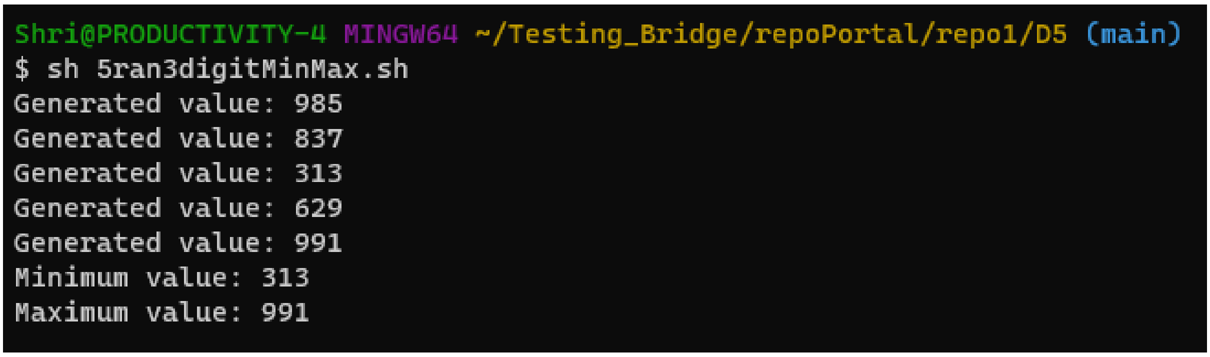
Selection Practice Problems with if & else

1. Write a program that reads 5 Random 3 Digit values and then outputs the minimum and the maximum value

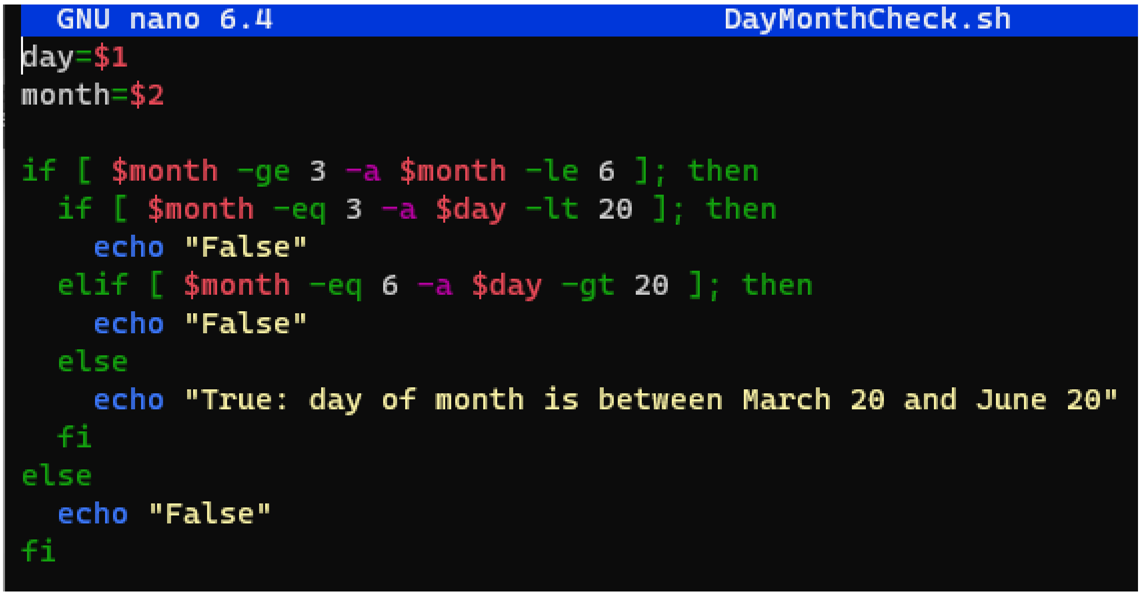


```
  GNU nano 6.4                            5ran3digitMinMax.sh
min_value=1000
max_value=0

for ((i = 1; i <= 5; i++)); do
    random_value=$((RANDOM % 900 + 100))

    echo "Generated value: $random_value"

    if [ $random_value -lt $min_value ]; then
        min_value=$random_value
    fi

    if [ $random_value -gt $max_value ]; then
        max_value=$random_value
    fi
done

echo "Minimum value: $min_value"
echo "Maximum value: $max_value"
```

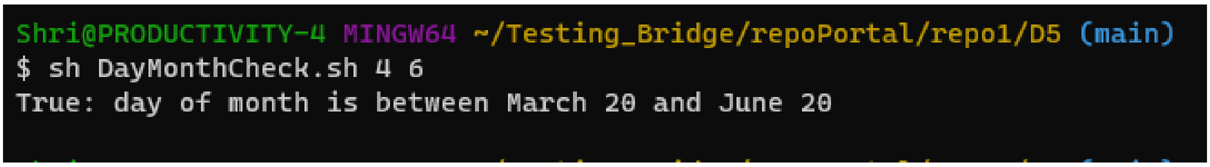


```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh 5ran3digitMinMax.sh
Generated value: 985
Generated value: 837
Generated value: 313
Generated value: 629
Generated value: 991
Minimum value: 313
Maximum value: 991
```

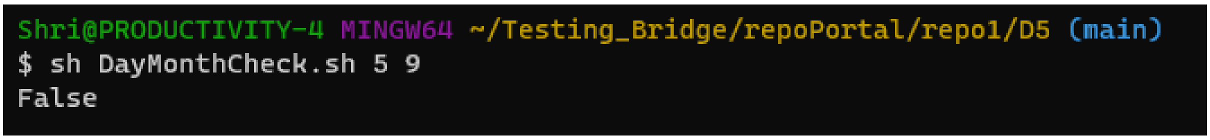
2. Write a program that takes day and month from the command line and prints true if day of month is between March 20 and June 20, false otherwise.



```
  GNU nano 6.4                        DayMonthCheck.sh
day=$1
month=$2

if [ $month -ge 3 -a $month -le 6 ]; then
  if [ $month -eq 3 -a $day -lt 20 ]; then
    echo "False"
  elif [ $month -eq 6 -a $day -gt 20 ]; then
    echo "False"
  else
    echo "True: day of month is between March 20 and June 20"
  fi
else
  echo "False"
fi
```

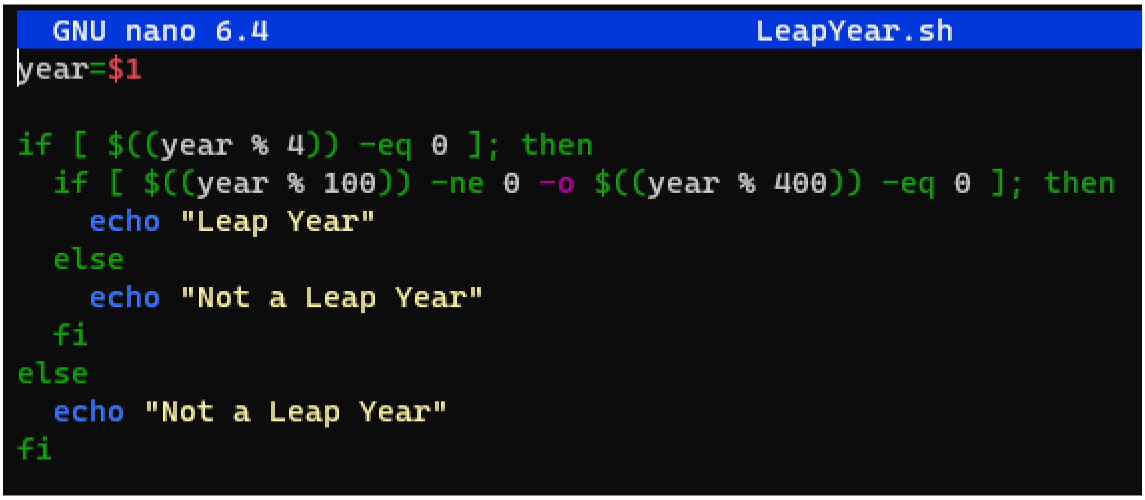


```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh DayMonthCheck.sh 4 6
True: day of month is between March 20 and June 20
```

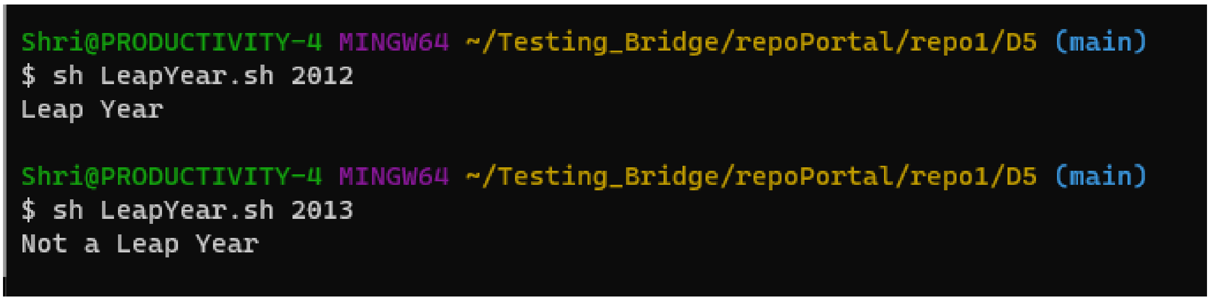


```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh DayMonthCheck.sh 5 9
False
```

3. Write a program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year. A Leap Year checks for 4 Digit Number, Divisible by 4 and not 100 unless divisible by 400.

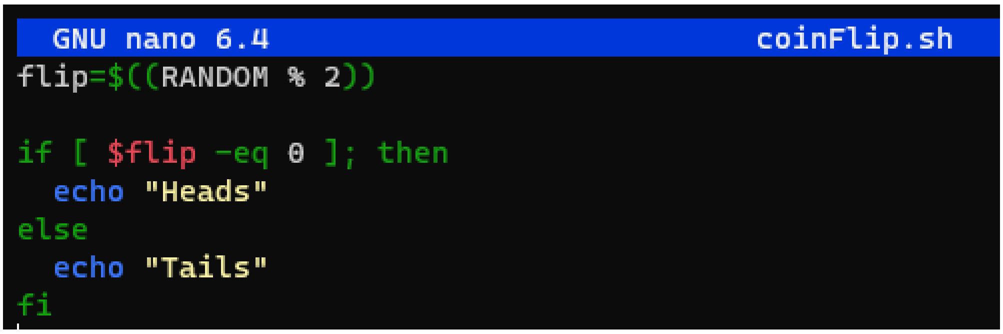


```
  GNU nano 6.4                          LeapYear.sh
year=$1

if [ $((year % 4)) -eq 0 ]; then
  if [ $((year % 100)) -ne 0 -o $((year % 400)) -eq 0 ]; then
    echo "Leap Year"
  else
    echo "Not a Leap Year"
  fi
else
  echo "Not a Leap Year"
fi
```

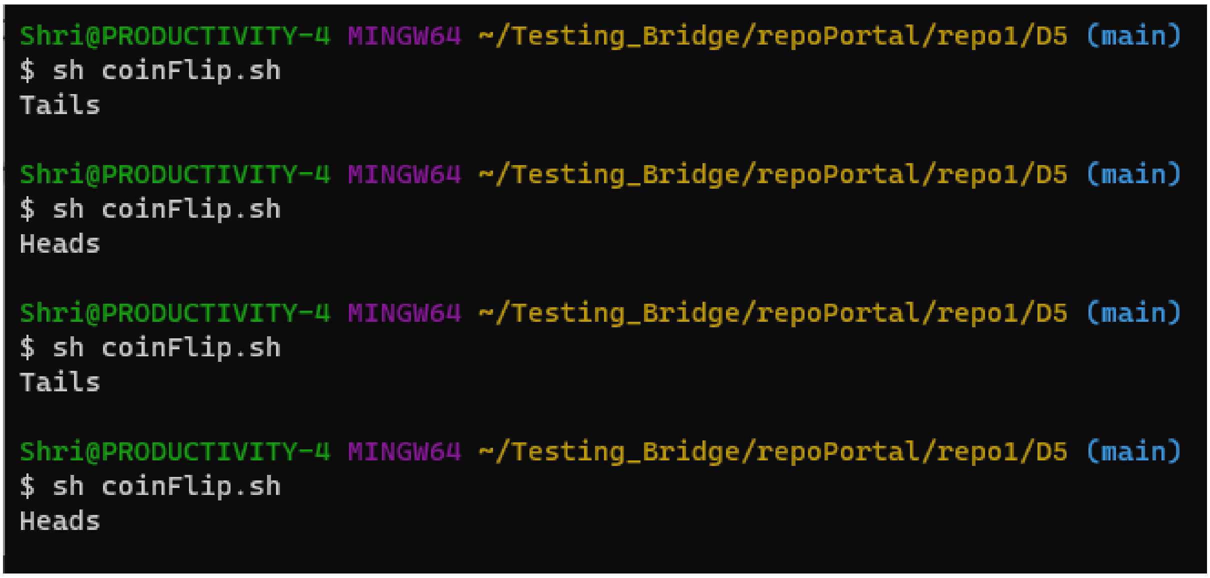


```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh LeapYear.sh 2012
Leap Year

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh LeapYear.sh 2013
Not a Leap Year
```

4. Write a program to simulate a coin flip and print out "Heads" or "Tails" accordingly.



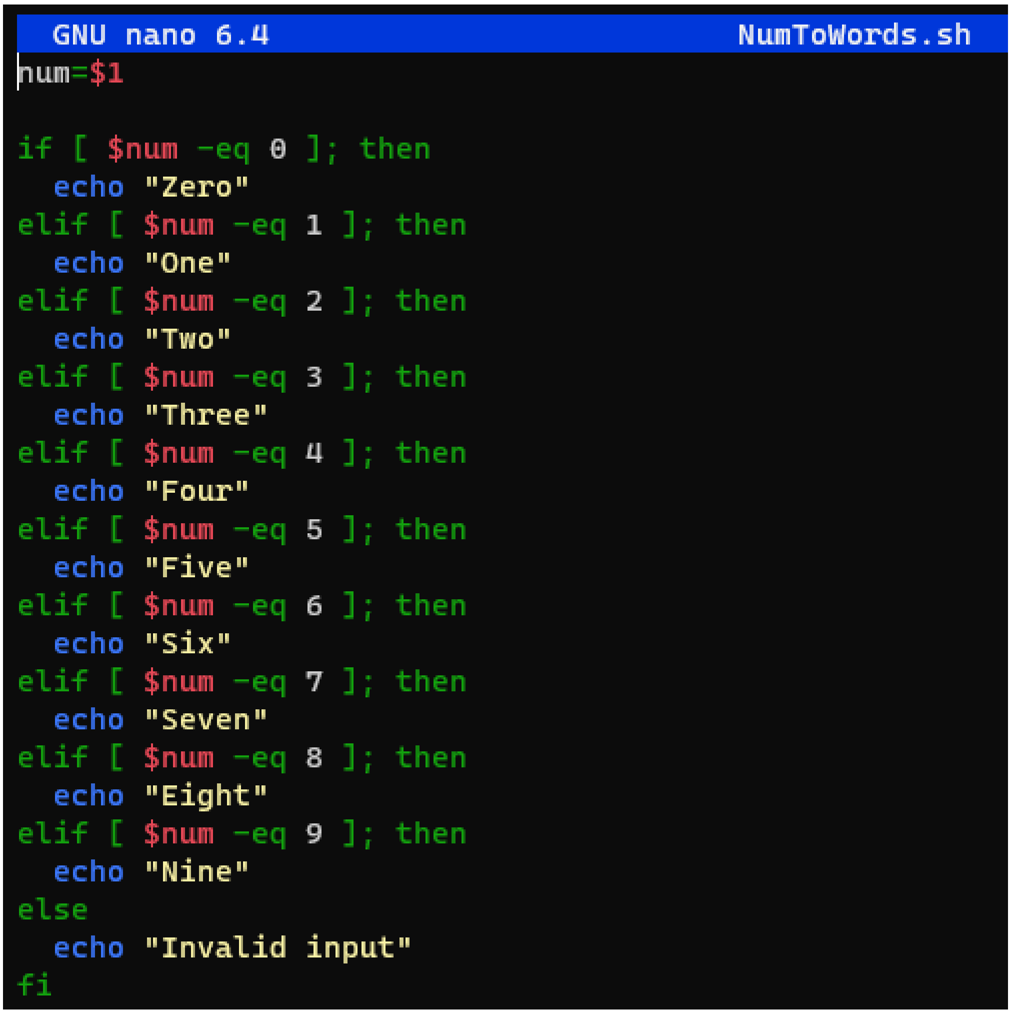
```
  GNU nano 6.4                          coinFlip.sh
flip=$((RANDOM % 2))

if [ $flip -eq 0 ]; then
  echo "Heads"
else
  echo "Tails"
fi
```



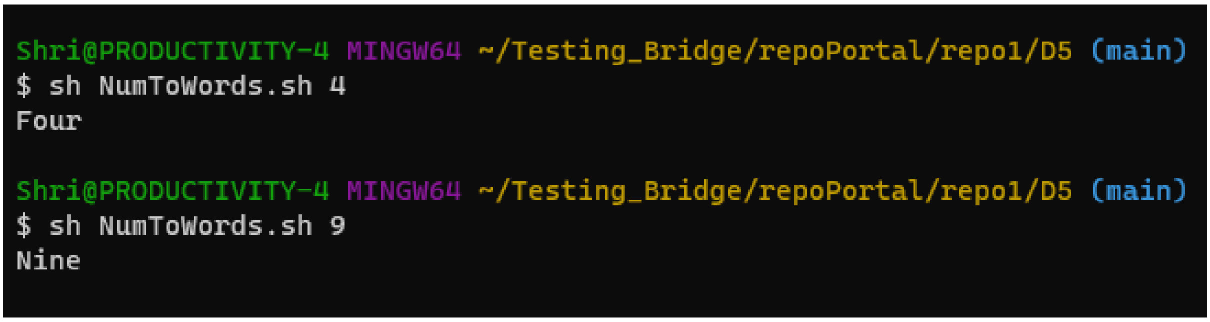
```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh coinFlip.sh
Tails

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh coinFlip.sh
Heads

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh coinFlip.sh
Tails

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh coinFlip.sh
Heads
```

Selection Practice Problems with if, elif and else

1. Read a single digit number and write the number in word

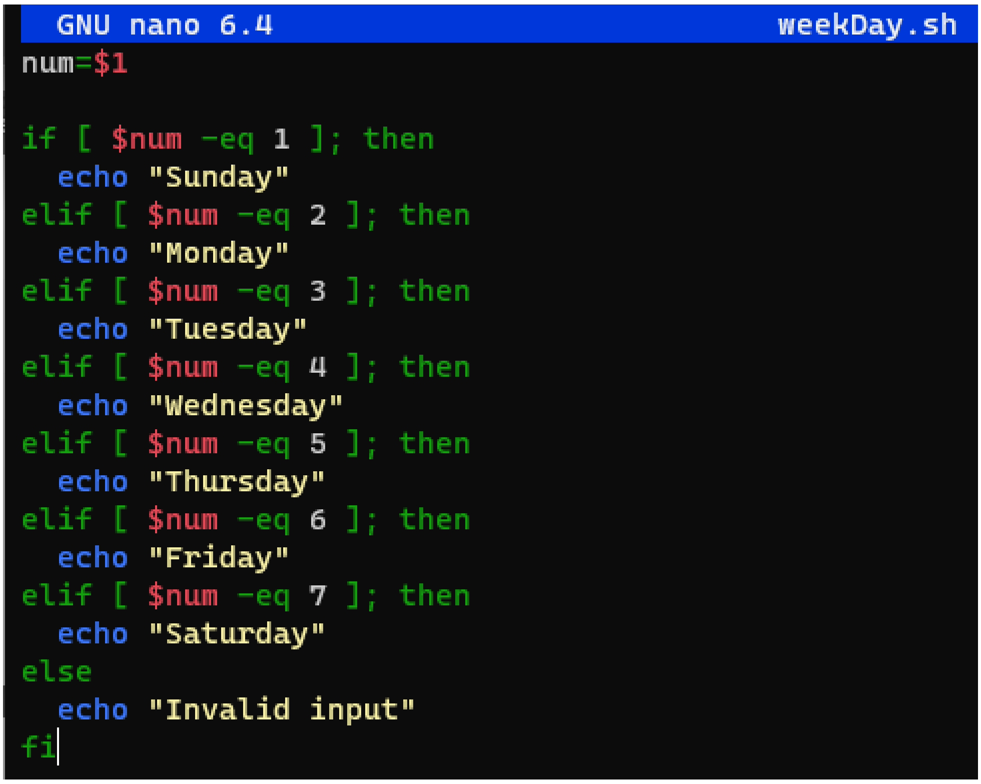


```
num=$1

if [ $num -eq 0 ]; then
  echo "Zero"
elif [ $num -eq 1 ]; then
  echo "One"
elif [ $num -eq 2 ]; then
  echo "Two"
elif [ $num -eq 3 ]; then
  echo "Three"
elif [ $num -eq 4 ]; then
  echo "Four"
elif [ $num -eq 5 ]; then
  echo "Five"
elif [ $num -eq 6 ]; then
  echo "Six"
elif [ $num -eq 7 ]; then
  echo "Seven"
elif [ $num -eq 8 ]; then
  echo "Eight"
elif [ $num -eq 9 ]; then
  echo "Nine"
else
  echo "Invalid input"
fi
```

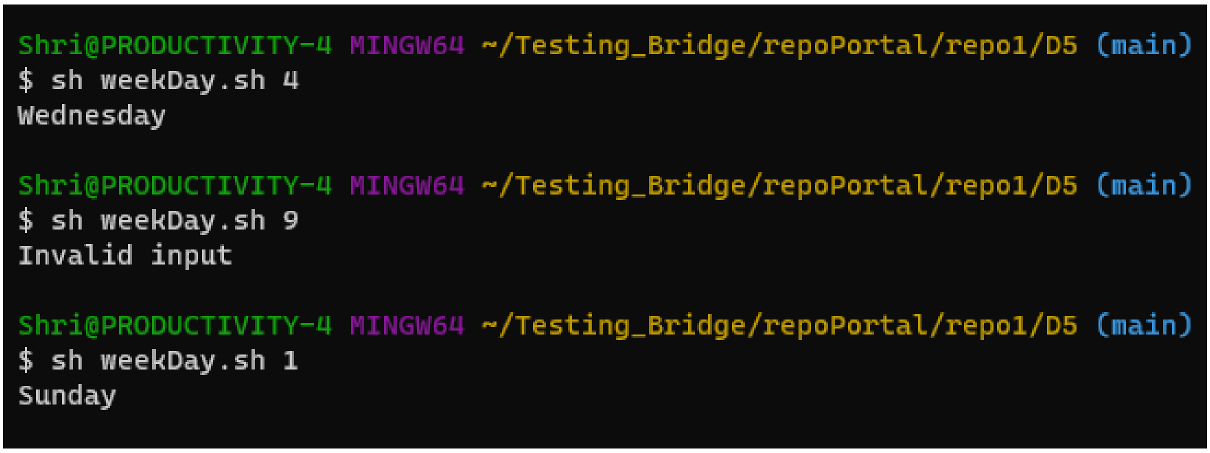


```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh NumToWords.sh 4
Four

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh NumToWords.sh 9
Nine
```

2. Read a Number and Display the week day (Sunday, Monday,...)

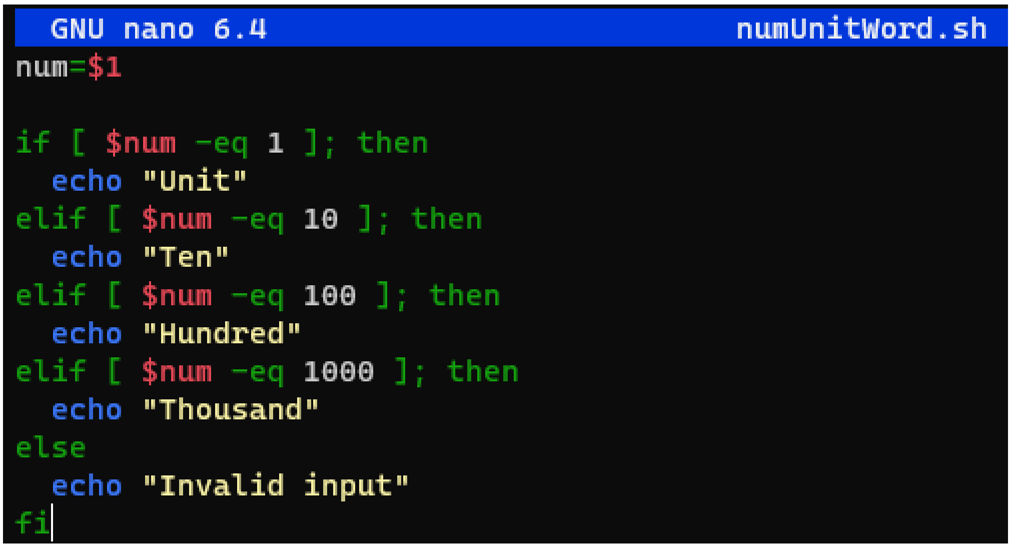


```
num=$1

if [ $num -eq 1 ]; then
  echo "Sunday"
elif [ $num -eq 2 ]; then
  echo "Monday"
elif [ $num -eq 3 ]; then
  echo "Tuesday"
elif [ $num -eq 4 ]; then
  echo "Wednesday"
elif [ $num -eq 5 ]; then
  echo "Thursday"
elif [ $num -eq 6 ]; then
  echo "Friday"
elif [ $num -eq 7 ]; then
  echo "Saturday"
else
  echo "Invalid input"
fi
```

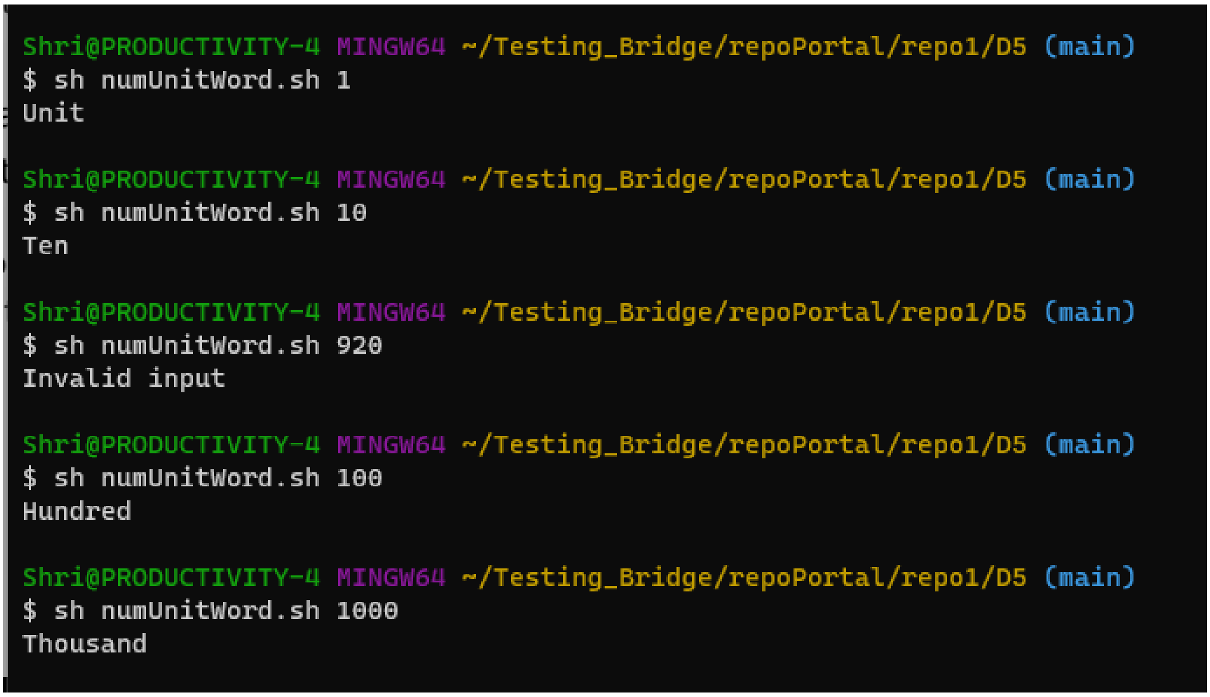


```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh weekDay.sh 4
Wednesday

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh weekDay.sh 9
Invalid input

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh weekDay.sh 1
Sunday
```

3. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,…

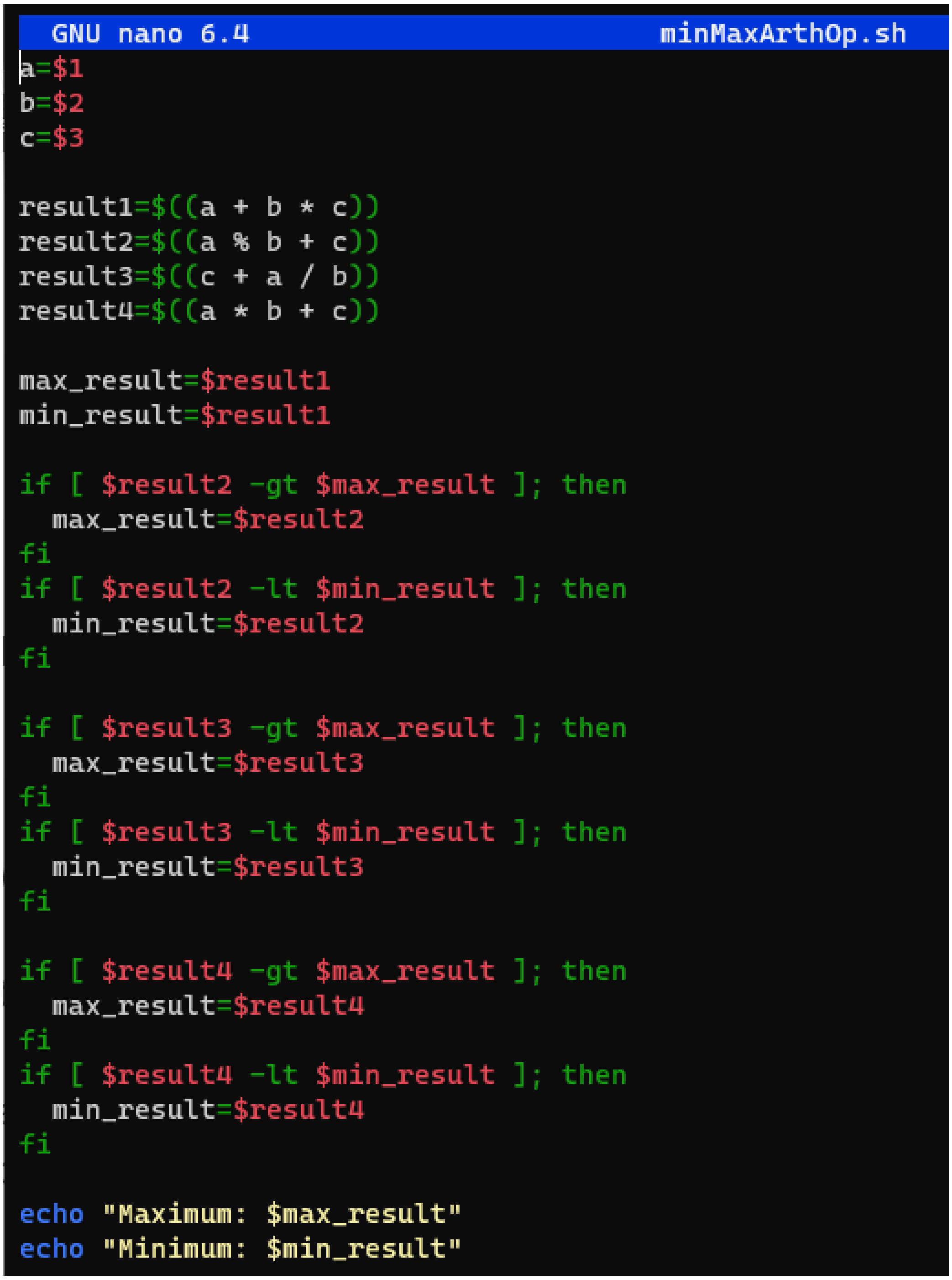


```
  GNU nano 6.4                                    numUnitWord.sh
num=$1

if [ $num -eq 1 ]; then
  echo "Unit"
elif [ $num -eq 10 ]; then
  echo "Ten"
elif [ $num -eq 100 ]; then
  echo "Hundred"
elif [ $num -eq 1000 ]; then
  echo "Thousand"
else
  echo "Invalid input"
fi
```

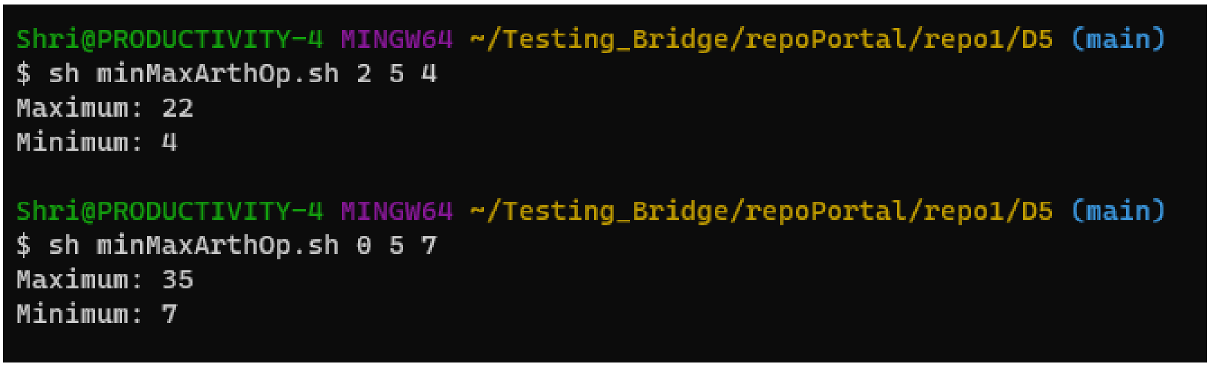


```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh numUnitWord.sh 1
Unit

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh numUnitWord.sh 10
Ten

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh numUnitWord.sh 920
Invalid input

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh numUnitWord.sh 100
Hundred

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh numUnitWord.sh 1000
Thousand
```

4. Enter 3 Numbers do following arithmetic operation and find the one that is maximum and minimum
    a. a + b * c
    b. a % b + c
    c. c + a / b
    d. a * b + c



```
  GNU nano 6.4                         minMaxArthOp.sh
a=$1
b=$2
c=$3

result1=$((a + b * c))
result2=$((a % b + c))
result3=$((c + a / b))
result4=$((a * b + c))

max_result=$result1
min_result=$result1

if [ $result2 -gt $max_result ]; then
  max_result=$result2
fi
if [ $result2 -lt $min_result ]; then
  min_result=$result2
fi

if [ $result3 -gt $max_result ]; then
  max_result=$result3
fi
if [ $result3 -lt $min_result ]; then
  min_result=$result3
fi

if [ $result4 -gt $max_result ]; then
  max_result=$result4
fi
if [ $result4 -lt $min_result ]; then
  min_result=$result4
fi

echo "Maximum: $max_result"
echo "Minimum: $min_result"
```



```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh minMaxArthOp.sh 2 5 4
Maximum: 22
Minimum: 4

Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo1/D5 (main)
$ sh minMaxArthOp.sh 0 5 7
Maximum: 35
Minimum: 7
```

Selection Practice Problems with case statement

1. Read a single digit number and write the number in word using Case
2. Read a Number and Display the week day (Sunday, Monday,...)
3. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,...
4. Write a program that takes User Inputs and does Unit Conversion of different Length units
    a. Feet to Inch
    b. Feet to Meter
    c. Inch to Feet
    d. Meter to Feet