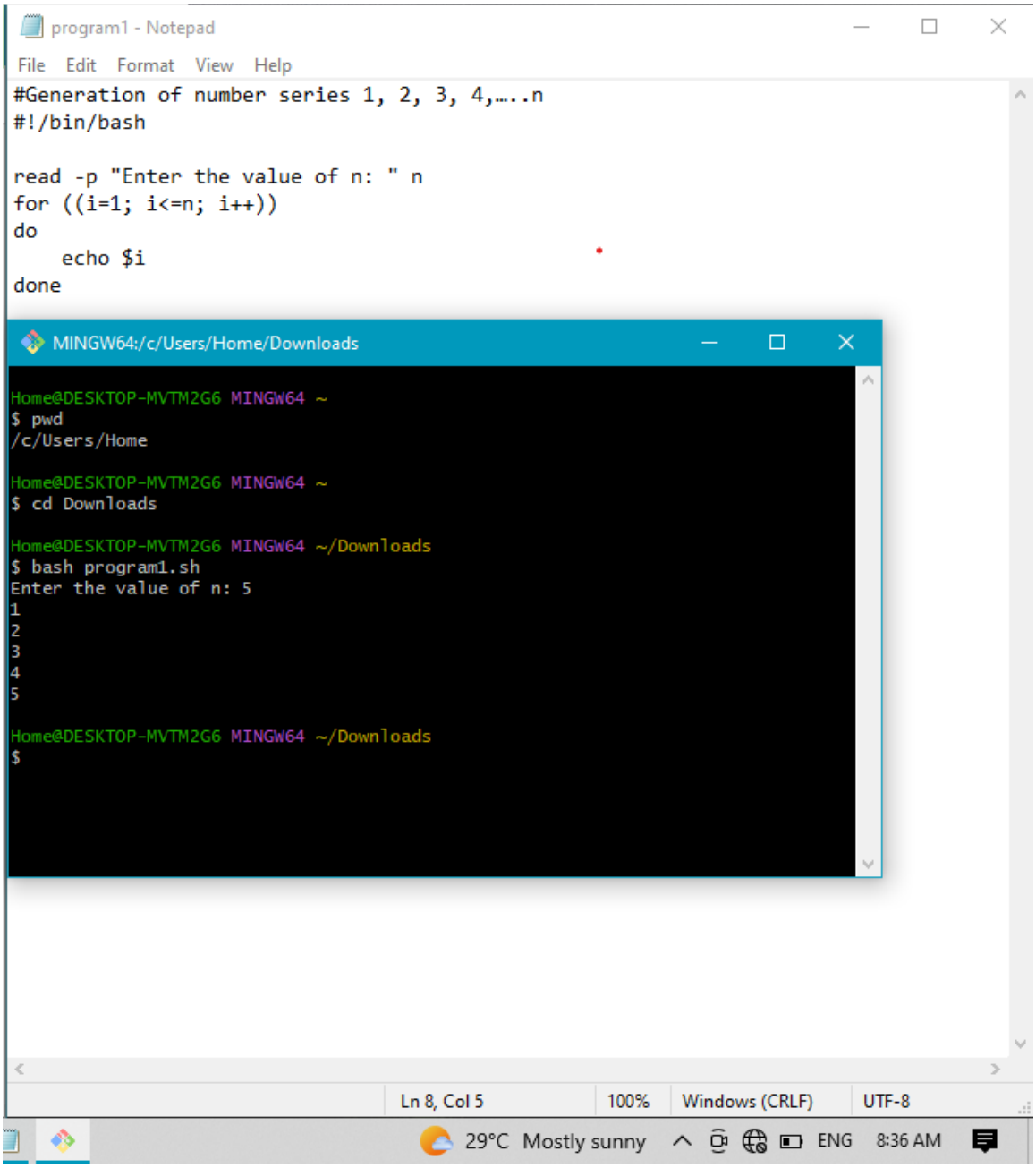
SHELL PROGRAMS

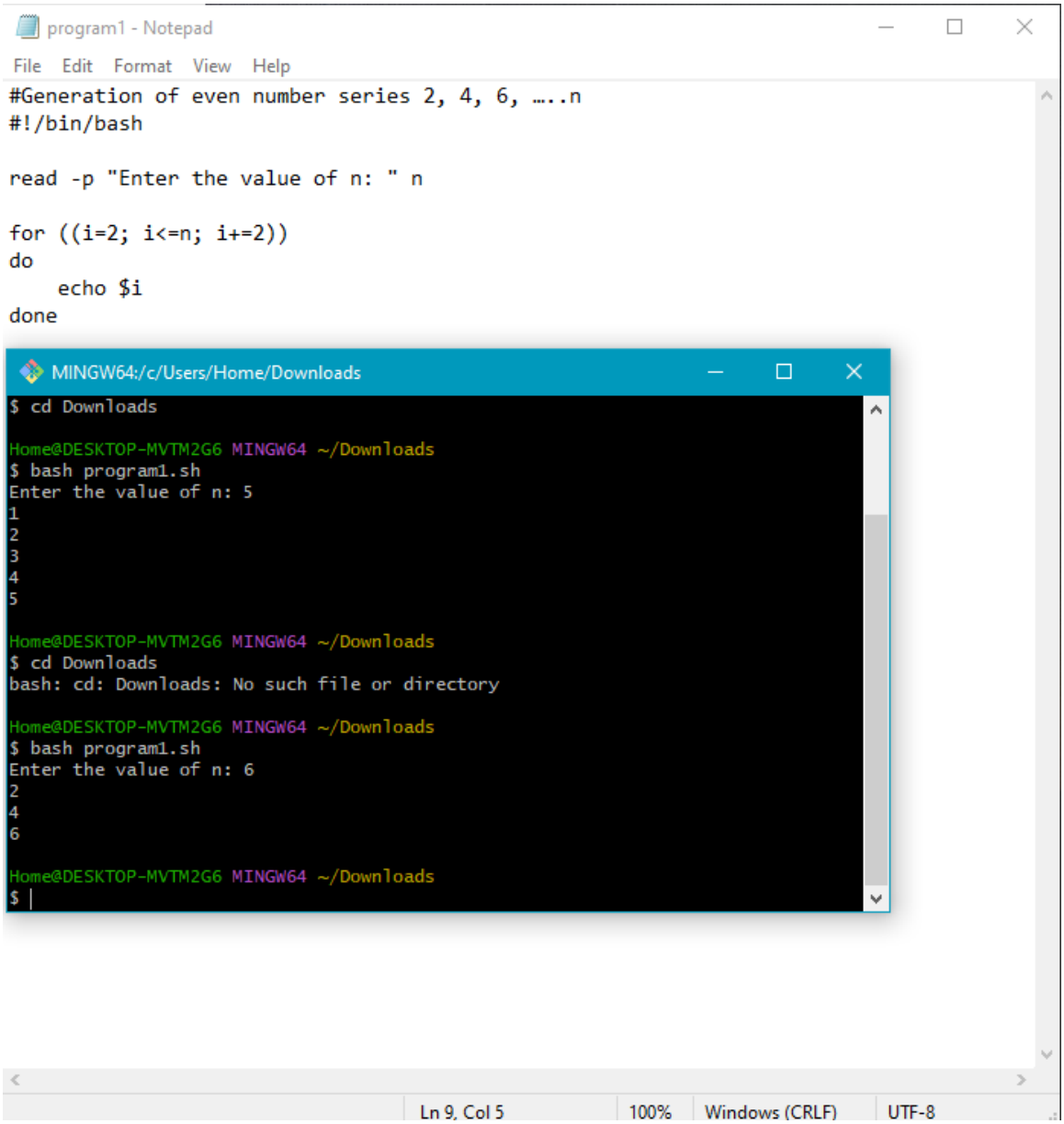
PROGRAM 1

CODE:



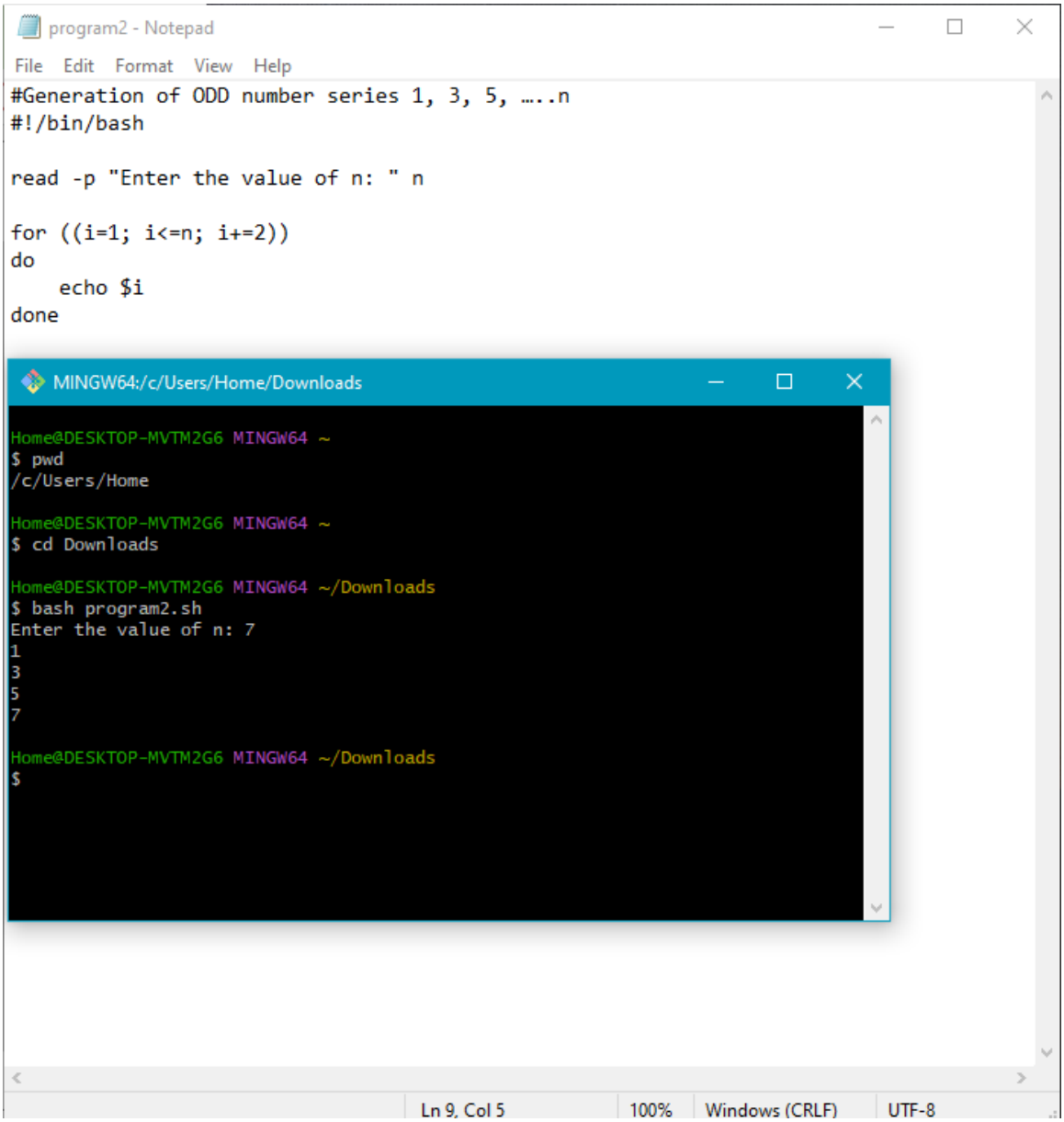
program 2:

CODE:



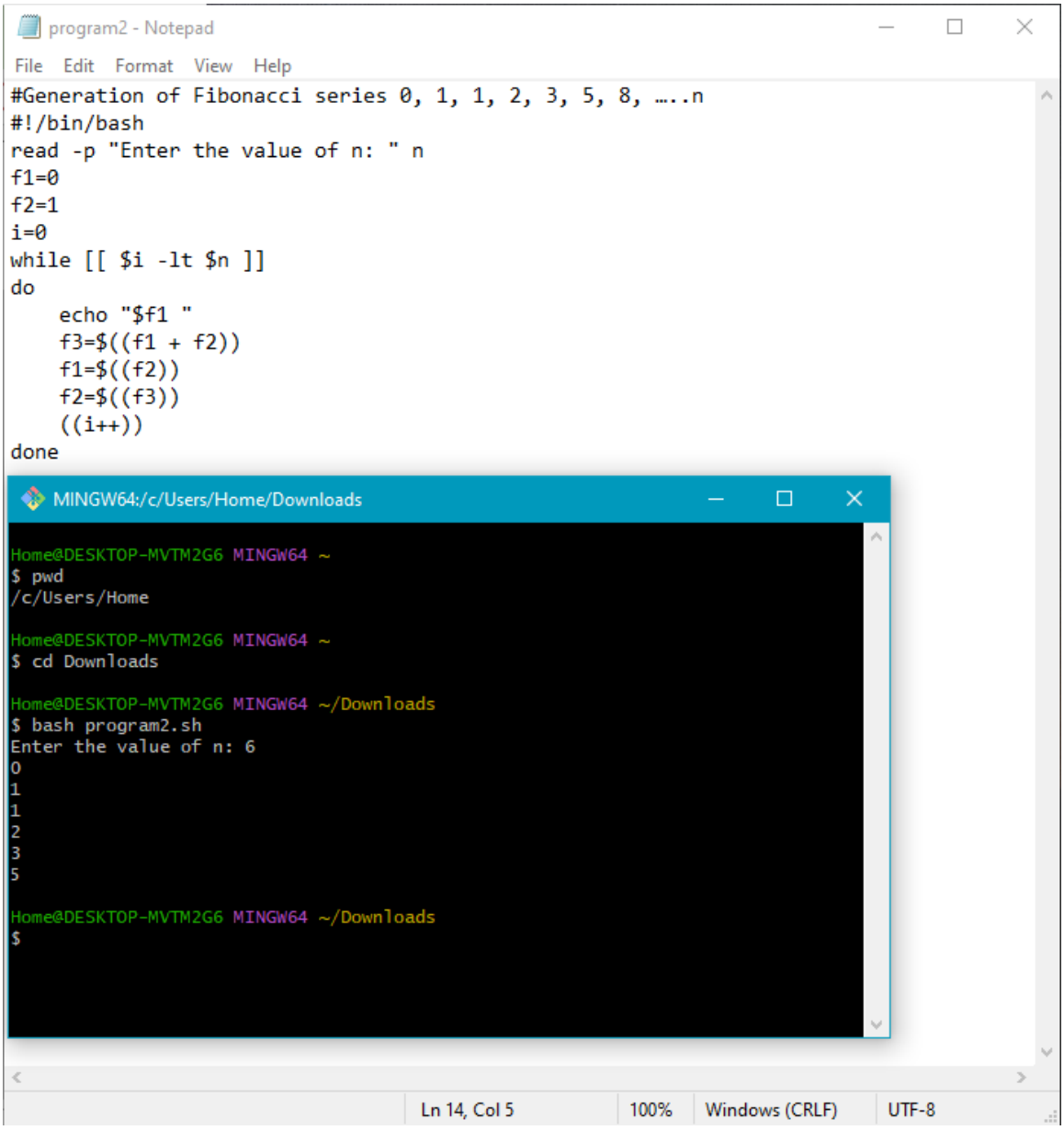
PROGRAM 3:

CODE:



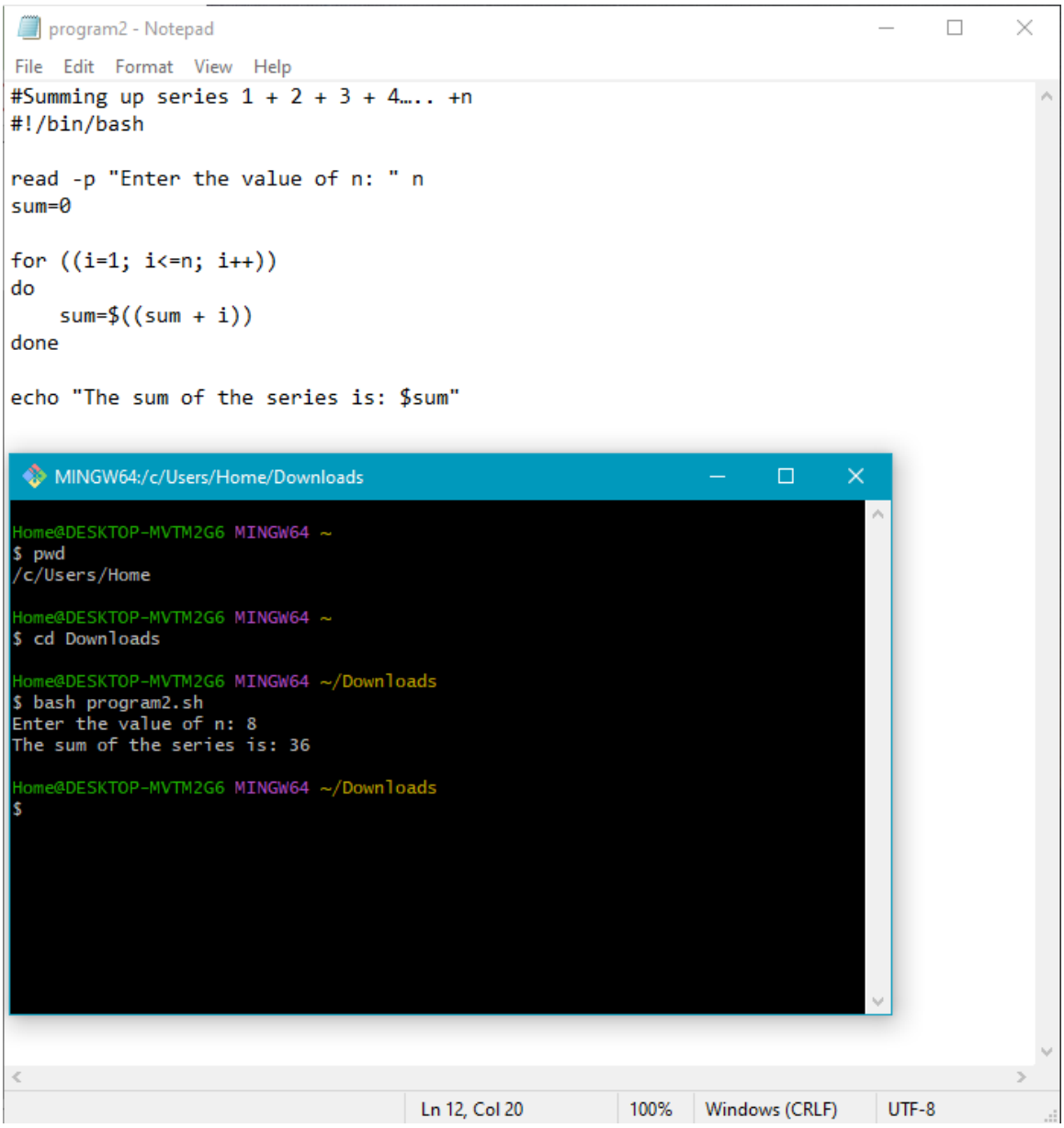
PROGRAM 4:

CODE:



PROGRAM 5:

CODE:



PROGRAM 6:

CODE:

pro3 - Notepad

File Edit Format View Help

```
#Summing up Even Number series
#!/bin/bash

read -p "Enter the value of n: " n
sum=0

for ((i=2; i<=n; i+=2))
do
    sum=$((sum + i))
done

echo "The sum of the even number series is: $sum"
```

MINGW64:/c/Users/Home/Downloads

```
Home@DESKTOP-MVTM2G6 MINGW64 ~
$ pwd
/c/Users/Home

Home@DESKTOP-MVTM2G6 MINGW64 ~
$ cd Downloads

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Enter the value of n: 7
The sum of the even number series is: 12

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
```

Ln 9, Col 21 100% Windows (CRLF) UTF-8

PROGRAM 7:

CODE :

pro3 - Notepad

File Edit Format View Help

```
#Summing up Odd Number series
#!/bin/bash

read -p "Enter the value of n: " n
sum=0

for ((i=1; i<=n; i+=2))
do
    sum=$((sum + i))
done

echo "The sum of the odd number series is: $sum"
```

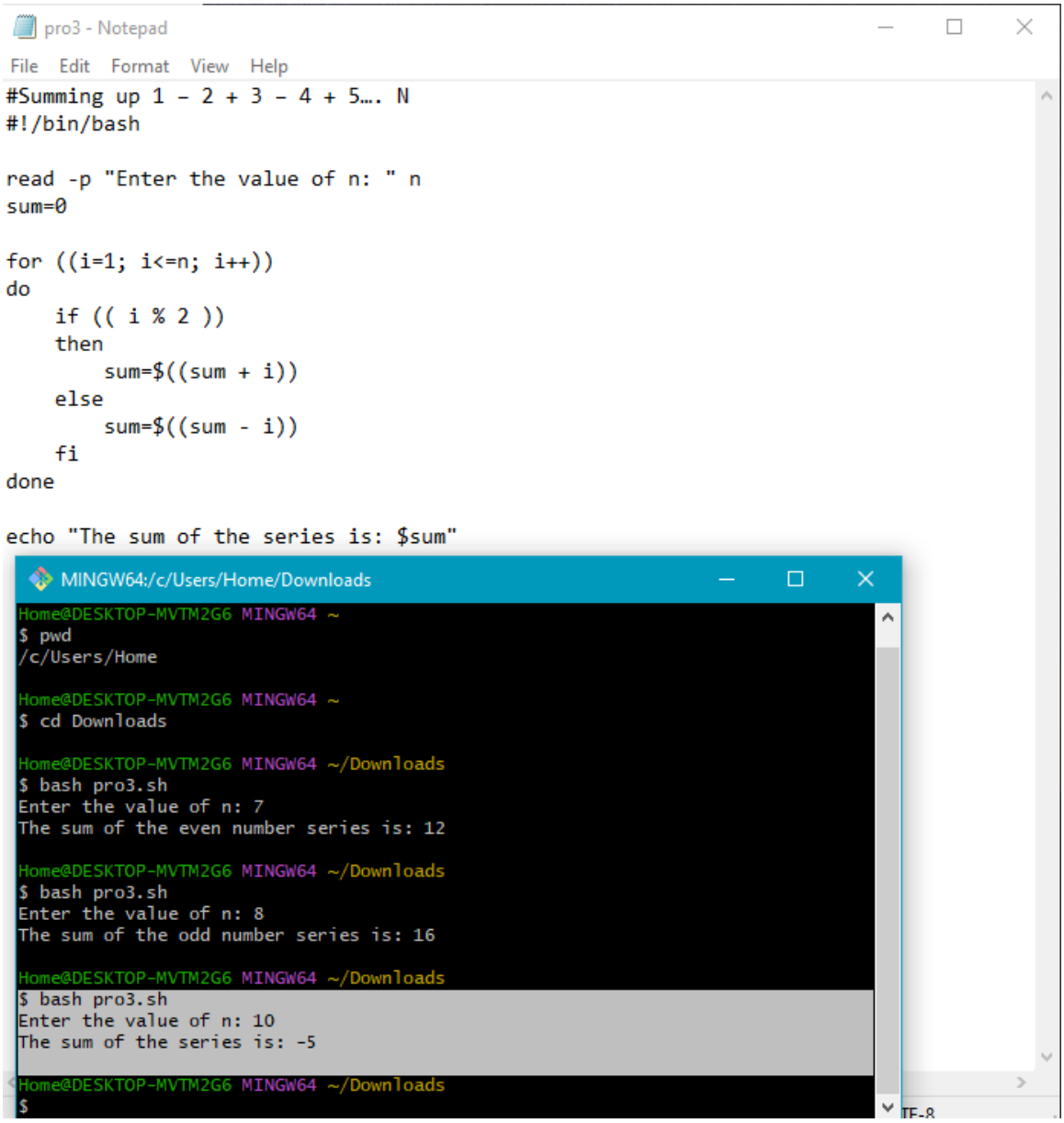
MINGW64:/c/Users/Home/Downloads

```
Home@DESKTOP-MVTM2G6 MINGW64 ~
$ pwd
/c/Users/Home

Home@DESKTOP-MVTM2G6 MINGW64 ~
$ cd Downloads

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Enter the value of n: 7
The sum of the even number series is: 12

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Enter the value of n: 8
The sum of the odd number series is: 16

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
```

Ln 12, Col 49 100% Windows (CRLF) UTF-8

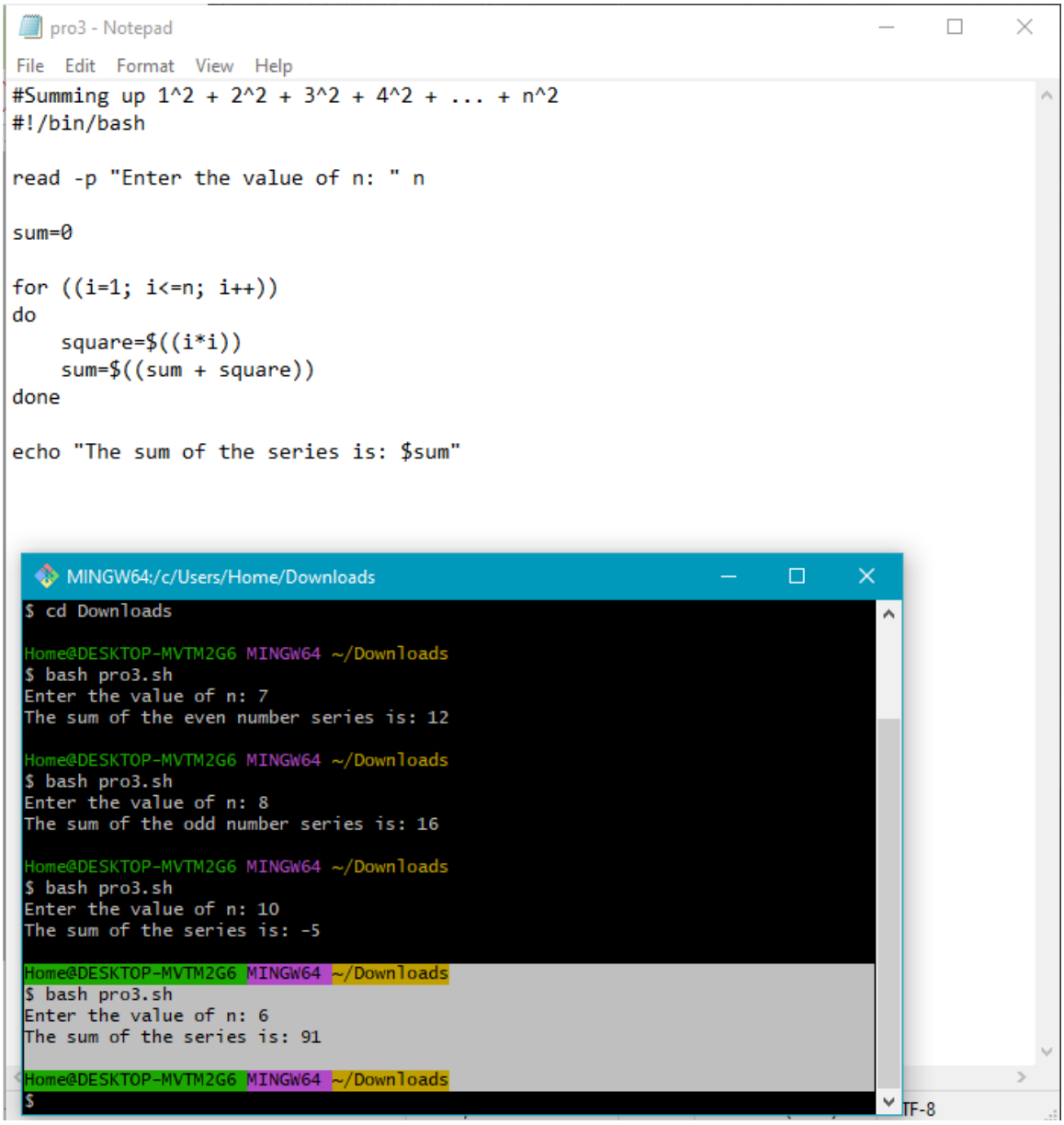
PROGRAM 8:

CODE :



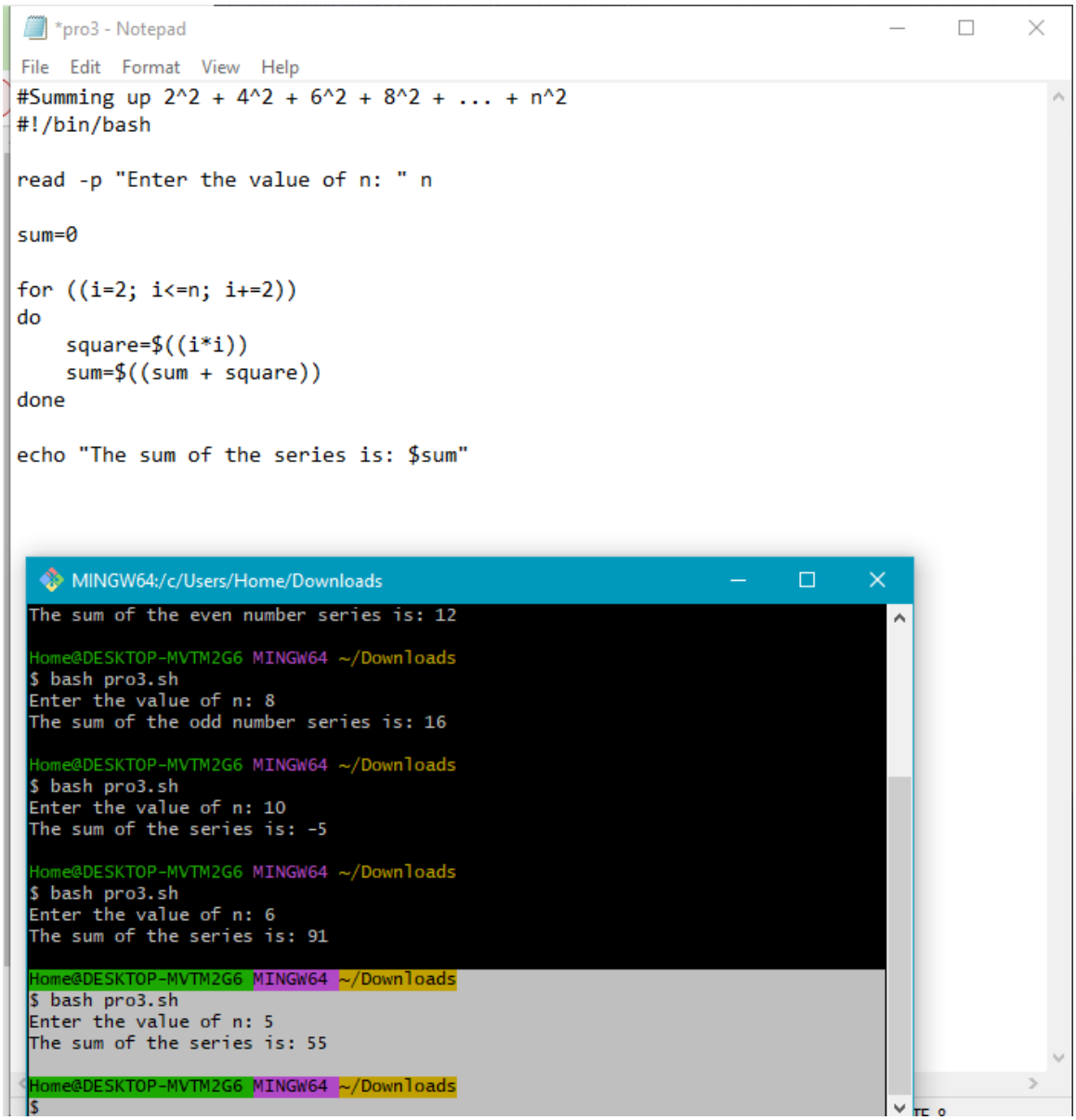
PROGRAM 9:

CODE:



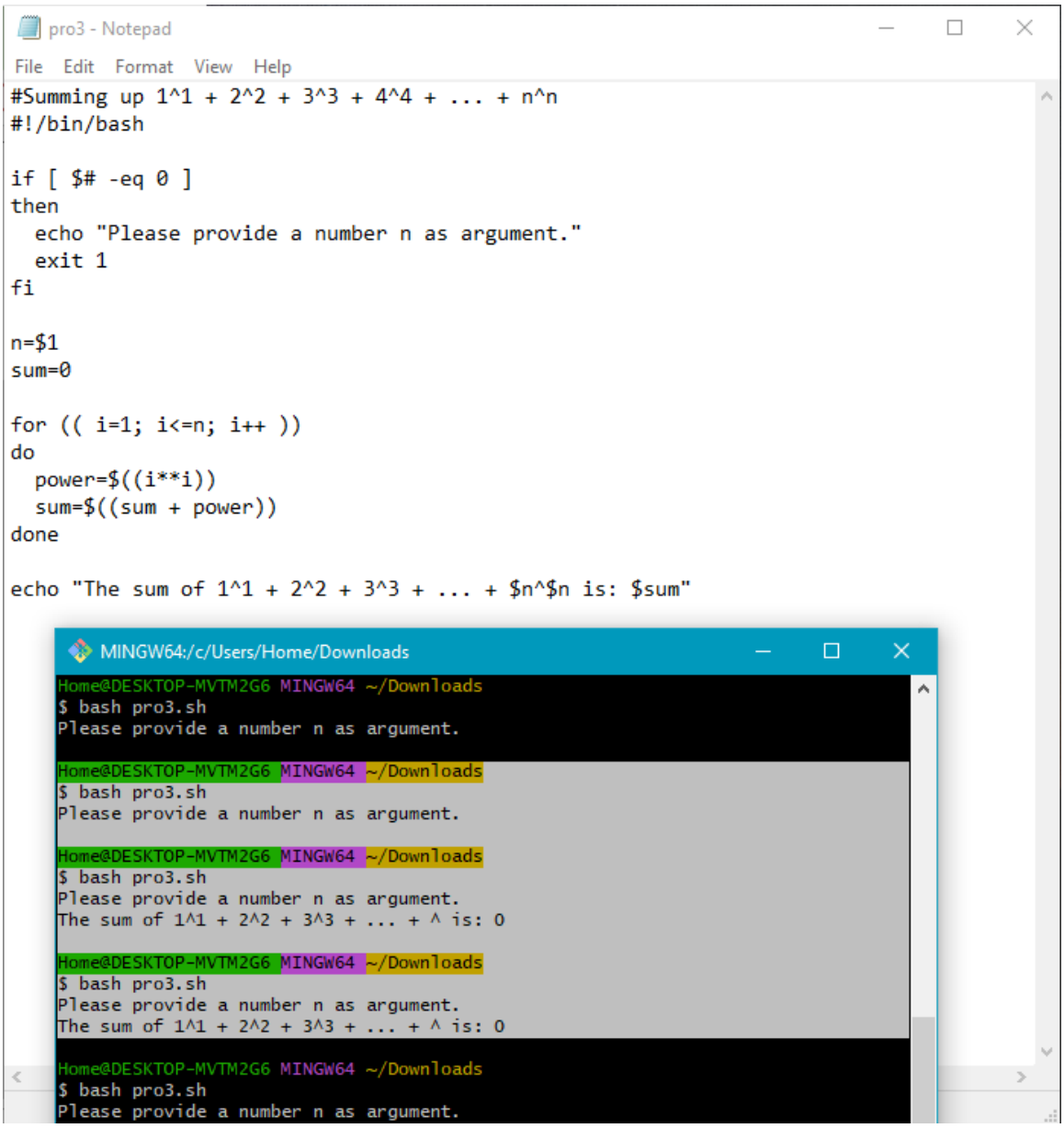
PROGRAM 10:

CODE :



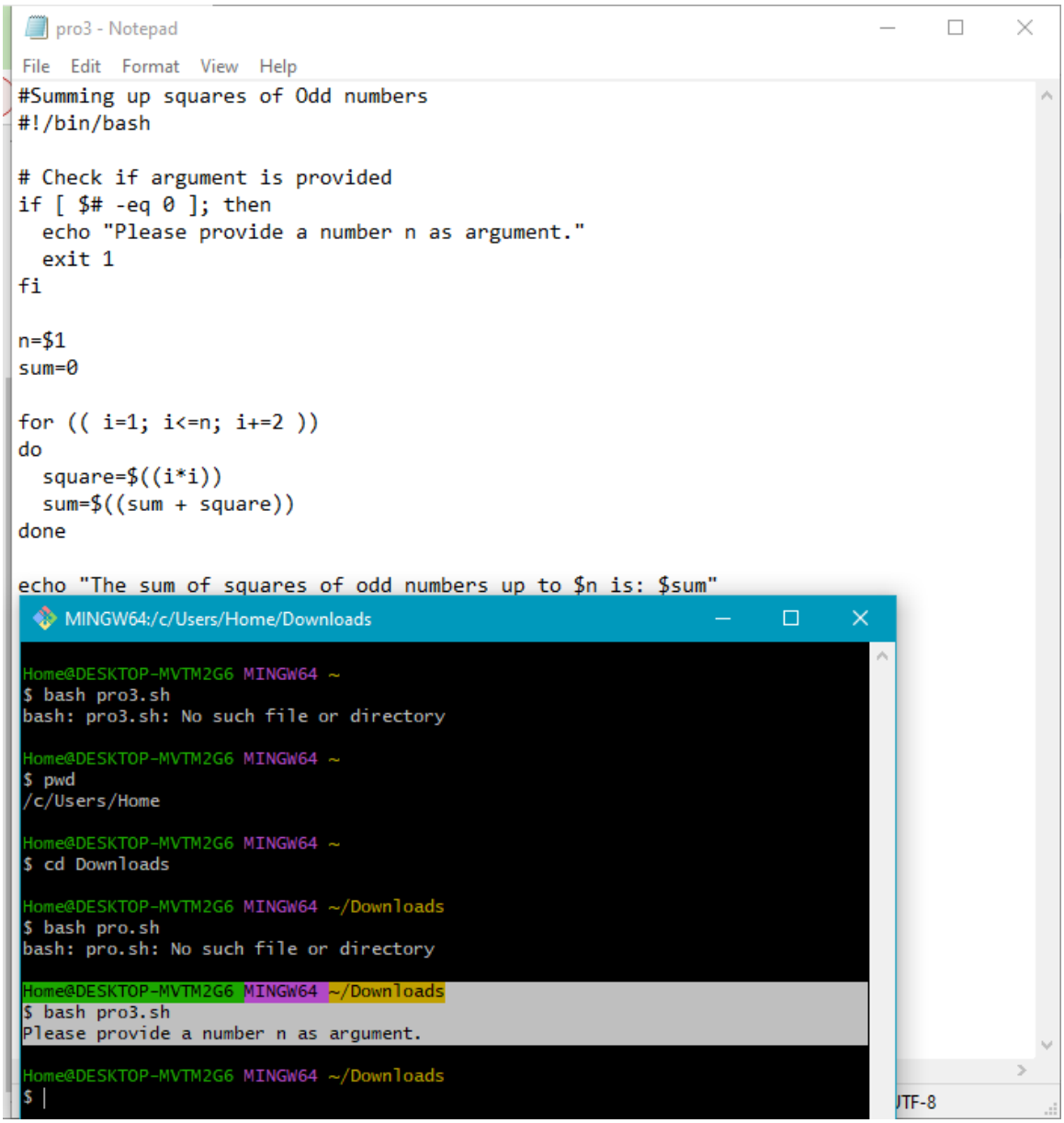
PROGRAM 11:

CODE :


pro3 - Notepad
File Edit Format View Help
#Summing up 1^1 + 2^2 + 3^3 + 4^4 + ... + n^n
#!/bin/bash
if [$# -eq 0]
then
echo "Please provide a number n as argument."
exit 1
fi
n=$1
sum=0
for ((i=1; i<=n; i++))
do
power=$((i**i))
sum=$((sum + power))
done
echo "The sum of 1^1 + 2^2 + 3^3 + ... + $n^$n is: $sum"
MINGW64:/c/Users/Home/Downloads
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Please provide a number n as argument.
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Please provide a number n as argument.
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Please provide a number n as argument.
The sum of 1^1 + 2^2 + 3^3 + ... + ^ is: 0
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Please provide a number n as argument.
The sum of 1^1 + 2^2 + 3^3 + ... + ^ is: 0
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Please provide a number n as argument.

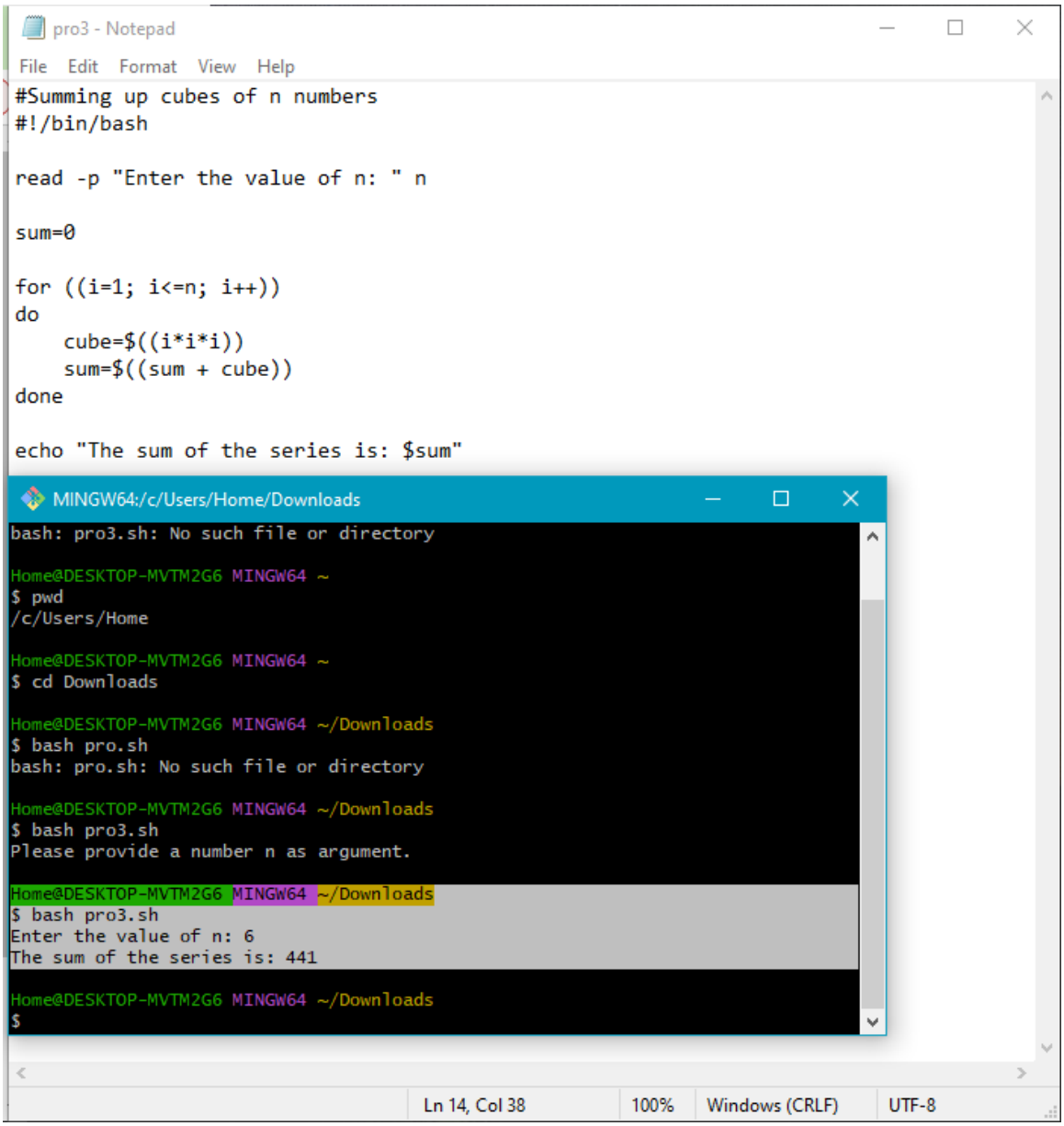
PROGRAM 12:

CODE :



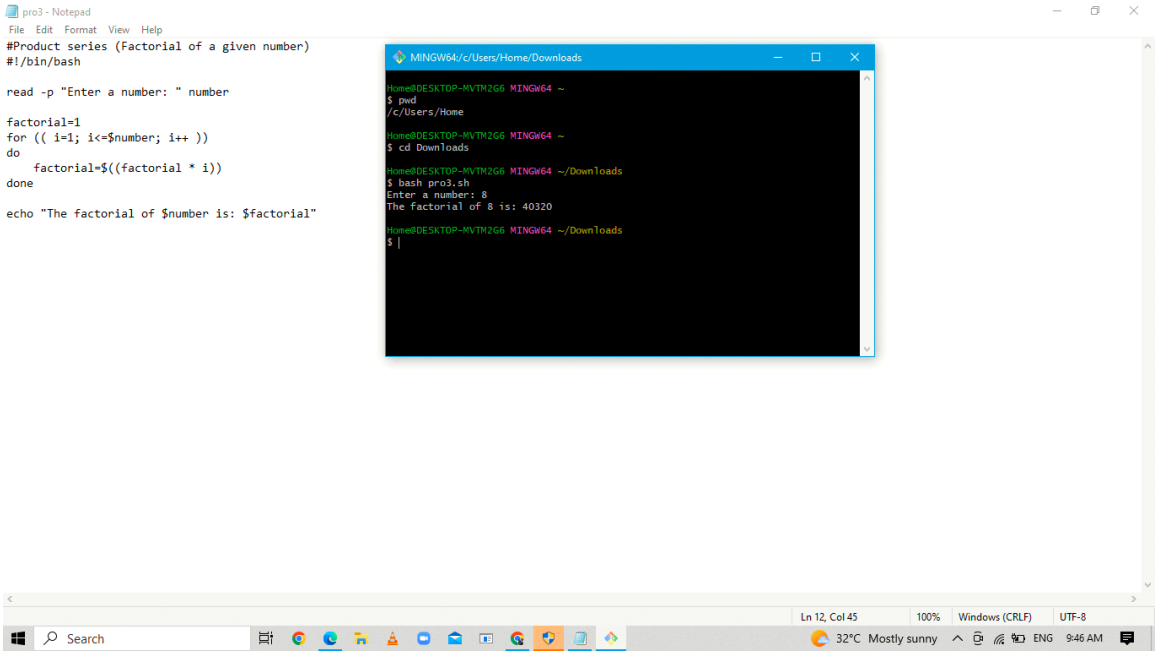
PROGRAM 13:

CODE :


pro3 - Notepad
File Edit Format View Help
#Summing up cubes of n numbers
#!/bin/bash
read -p "Enter the value of n: " n
sum=0
for ((i=1; i<=n; i++))
do
cube=$((i*i*i))
sum=$((sum + cube))
done
echo "The sum of the series is: $sum"
MINGW64:/c/Users/Home/Downloads
bash: pro3.sh: No such file or directory
Home@DESKTOP-MVTM2G6 MINGW64 ~
$ pwd
/c/Users/Home
Home@DESKTOP-MVTM2G6 MINGW64 ~
$ cd Downloads
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro.sh
bash: pro.sh: No such file or directory
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Please provide a number n as argument.
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash pro3.sh
Enter the value of n: 6
The sum of the series is: 441
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
Ln 14, Col 38
100%
Windows (CRLF)
UTF-8

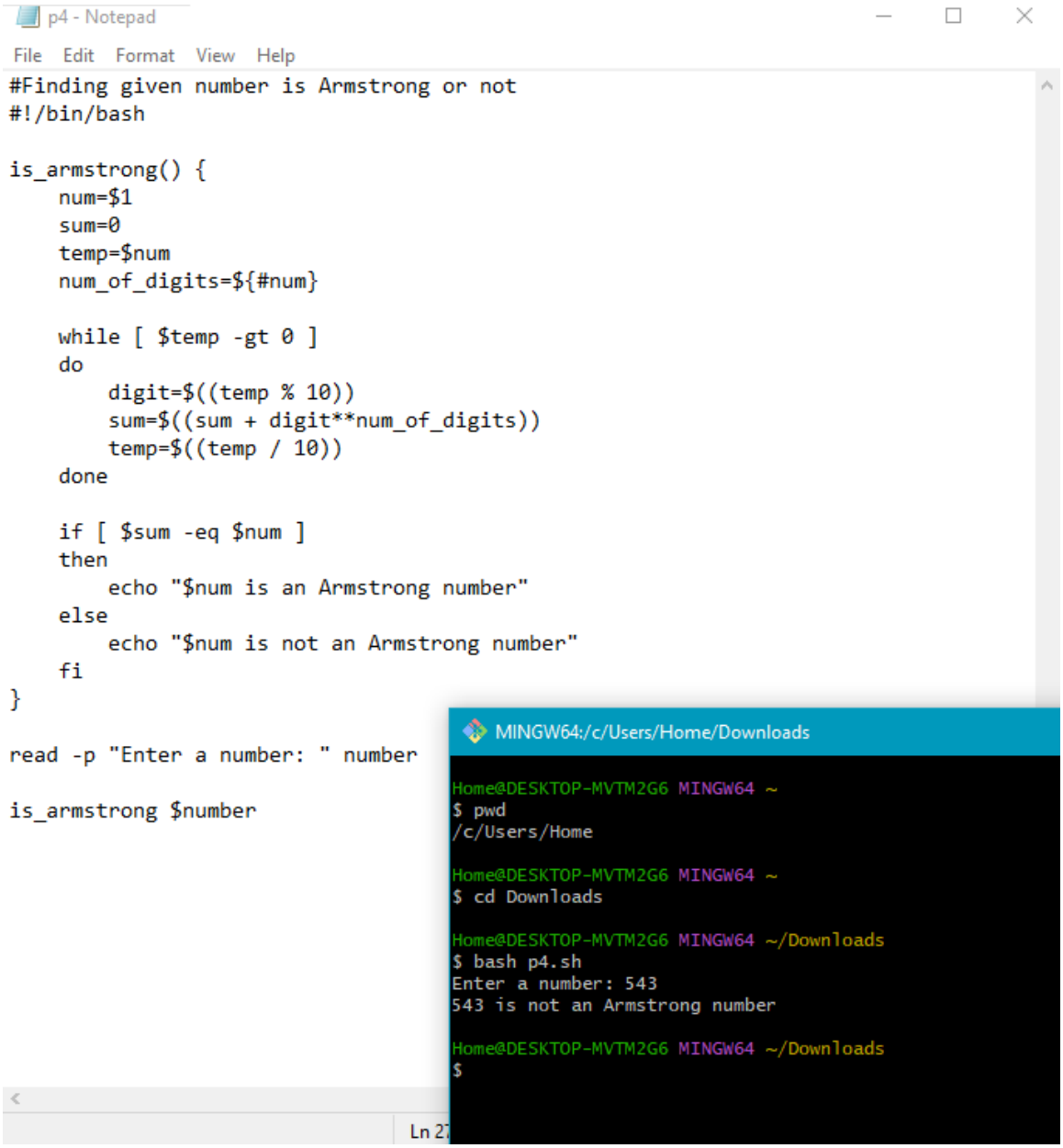
PROGRAM 14:

CODE :



PROGRAM 15:

CODE :


p4 - Notepad
File Edit Format View Help
#Finding given number is Armstrong or not
#!/bin/bash
is_armstrong() {
num=$1
sum=0
temp=$num
num_of_digits=${#num}
while [$temp -gt 0]
do
digit=$((temp % 10))
sum=$((sum + digit**num_of_digits))
temp=$((temp / 10))
done
if [$sum -eq $num]
then
echo "$num is an Armstrong number"
else
echo "$num is not an Armstrong number"
fi
}
read -p "Enter a number: " number
is_armstrong $number
MINGW64:/c/Users/Home/Downloads
Home@DESKTOP-MVTM2G6 MINGW64 ~
$ pwd
/c/Users/Home
Home@DESKTOP-MVTM2G6 MINGW64 ~
$ cd Downloads
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter a number: 543
543 is not an Armstrong number
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$

PROGRAM 16 :

CODE :

p4 - Notepad

File Edit Format View Help

```
#Summing up any n numbers and finding average
#!/bin/bash

read -p "Enter the total count of numbers:" N

total_sum=0
count=0

for ((i=1; i<=N; i++))
do
    read -p "Enter number $i:" number

    total_sum=$((total_sum + number))
done

average=$(echo "scale=2; $total_sum / $N" | bc)

echo "Sum: $total_sum"
echo "Average: $average"
```

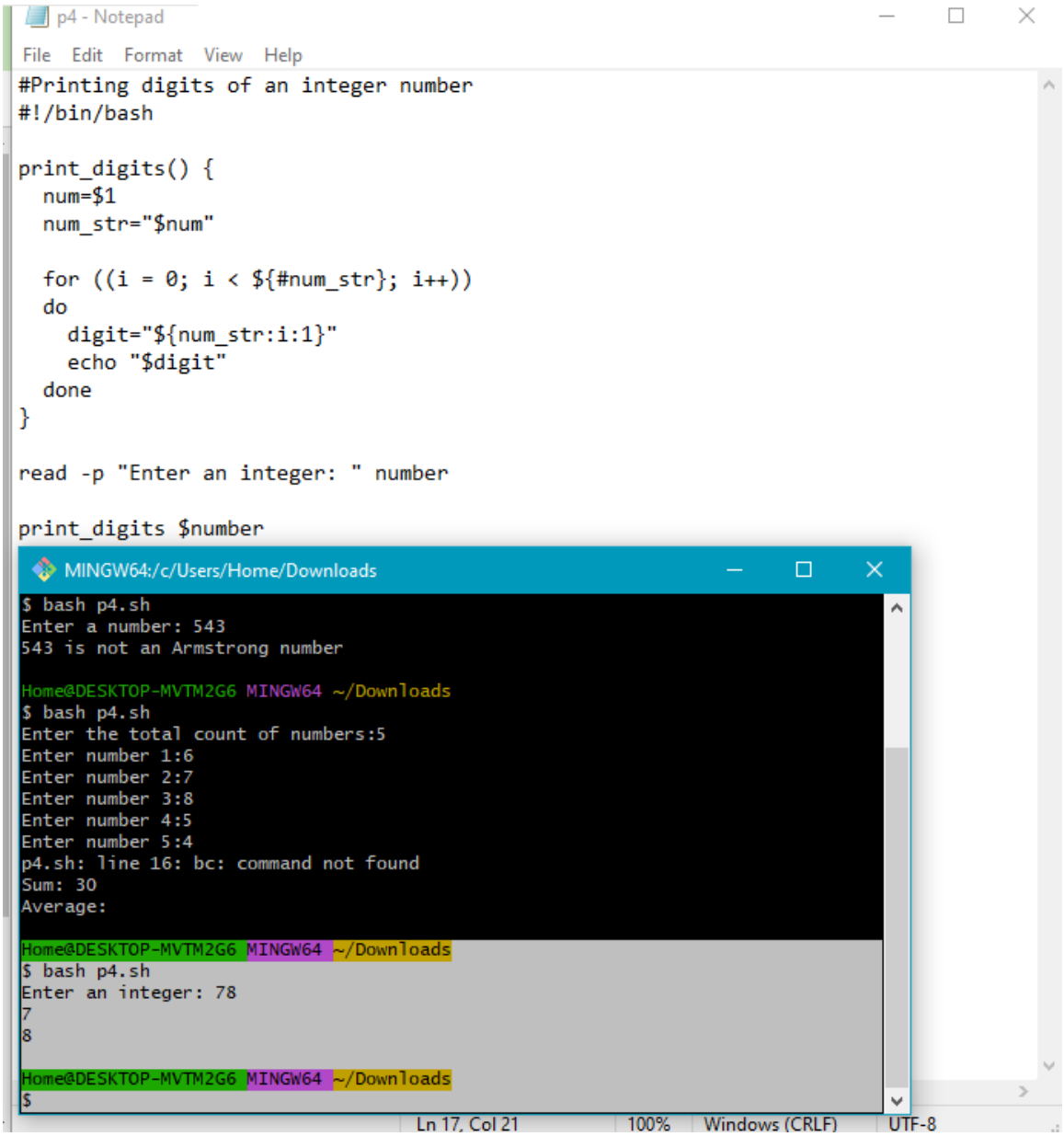
MINGW64:/c/Users/Home/Downloads

```
/c/Users/Home

Home@DESKTOP-MVTM2G6 MINGW64 ~
$ cd Downloads

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter a number: 543
543 is not an Armstrong number

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter the total count of numbers:5
Enter number 1:6
Enter number 2:7
Enter number 3:8
Enter number 4:5
Enter number 5:4
p4.sh: line 16: bc: command not found
Sum: 30
Average:
```

F-8

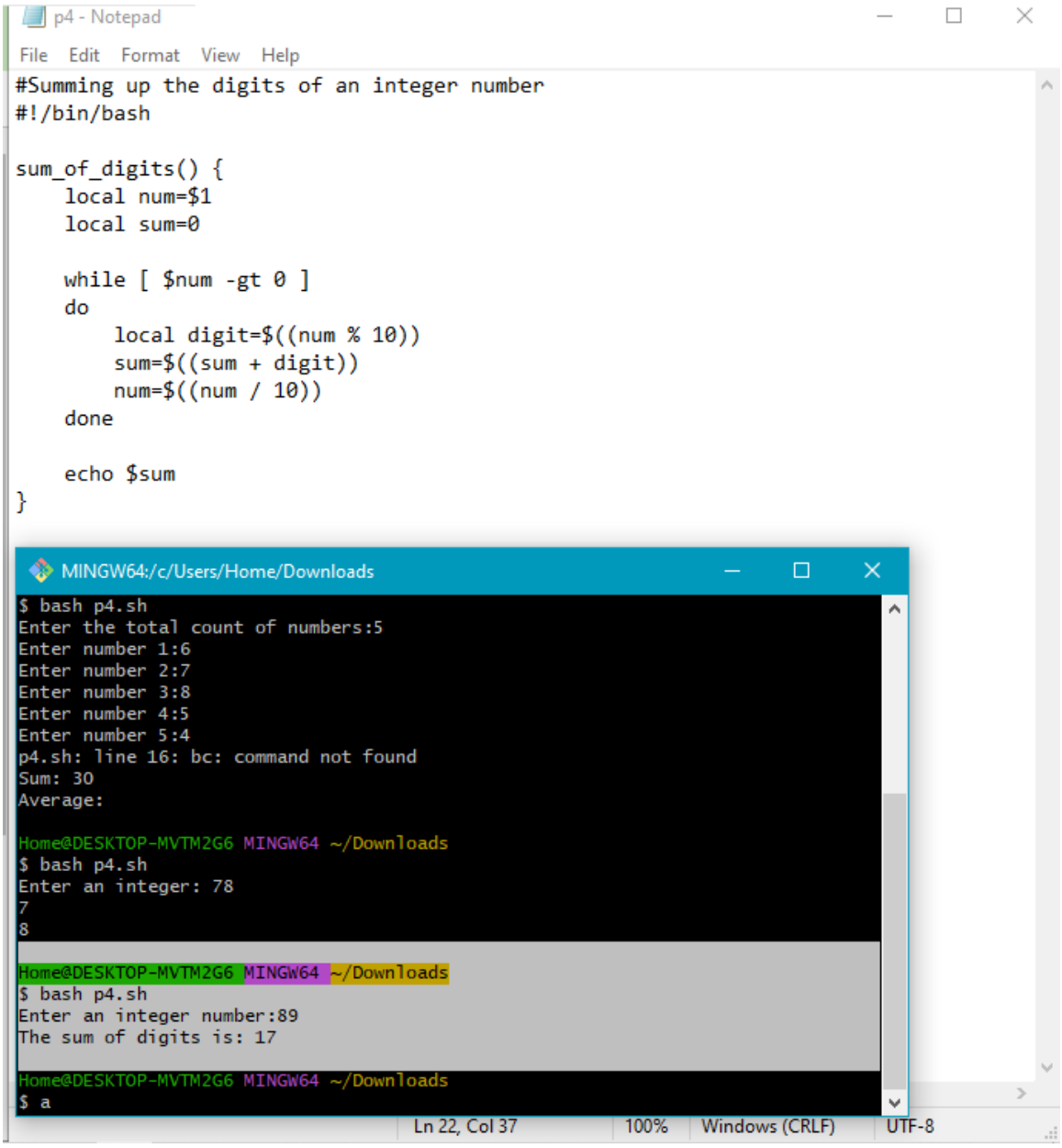
PROGRAM 17:

CODE :


p4 - Notepad
File Edit Format View Help
#Printing digits of an integer number
#!/bin/bash
print_digits() {
num=$1
num_str="$num"
for ((i = 0; i < ${#num_str}; i++))
do
digit="${num_str:i:1}"
echo "$digit"
done
}
read -p "Enter an integer: " number
print_digits $number
MINGW64:/c/Users/Home/Downloads
$ bash p4.sh
Enter a number: 543
543 is not an Armstrong number
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter the total count of numbers:5
Enter number 1:6
Enter number 2:7
Enter number 3:8
Enter number 4:5
Enter number 5:4
p4.sh: line 16: bc: command not found
Sum: 30
Average:
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer: 78
7
8
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
Ln 17, Col 21
100%
Windows (CRLF)
UTF-8

PROGRAM 18:

CODE:


p4 - Notepad
File Edit Format View Help
#Summing up the digits of an integer number
#!/bin/bash
sum_of_digits() {
local num=$1
local sum=0
while [$num -gt 0]
do
local digit=$((num % 10))
sum=$((sum + digit))
num=$((num / 10))
done
echo $sum
}
MINGW64:/c/Users/Home/Downloads
$ bash p4.sh
Enter the total count of numbers:5
Enter number 1:6
Enter number 2:7
Enter number 3:8
Enter number 4:5
Enter number 5:4
p4.sh: line 16: bc: command not found
Sum: 30
Average:
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer: 78
7
8
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number:89
The sum of digits is: 17
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ a
Ln 22, Col 37
100%
Windows (CRLF)
UTF-8

PROGRAM 19:

CODE :

p4 - Notepad

File Edit Format View Help

```
#Revering the digits of an integer number
#!/bin/bash

reverse_digits() {
  local number=$1
  local reversed=""

  while [ $number -gt 0 ]
  do
    local digit=$((number % 10))
    reversed="${reversed}${digit}"
    number=$((number / 10))
  done

  echo "$reversed"
}
```

MINGW64:/c/Users/Home/Downloads

```
Enter number 4:5
Enter number 5:4
p4.sh: line 16: bc: command not found
Sum: 30
Average:

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer: 78
7
8

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number:89
The sum of digits is: 17

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 567
Reversed number: 765

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
```

PROGRAM 20 :

CODE :

p4 - Notepad

File Edit Format View Help

```
#Finding whether the given integer is odd or even
#!/bin/bash

read -p "Enter an integer number: " input_number

if (( input_number % 2 ))
then
  echo "The number is odd."
else
  echo "The number is even."
fi
```

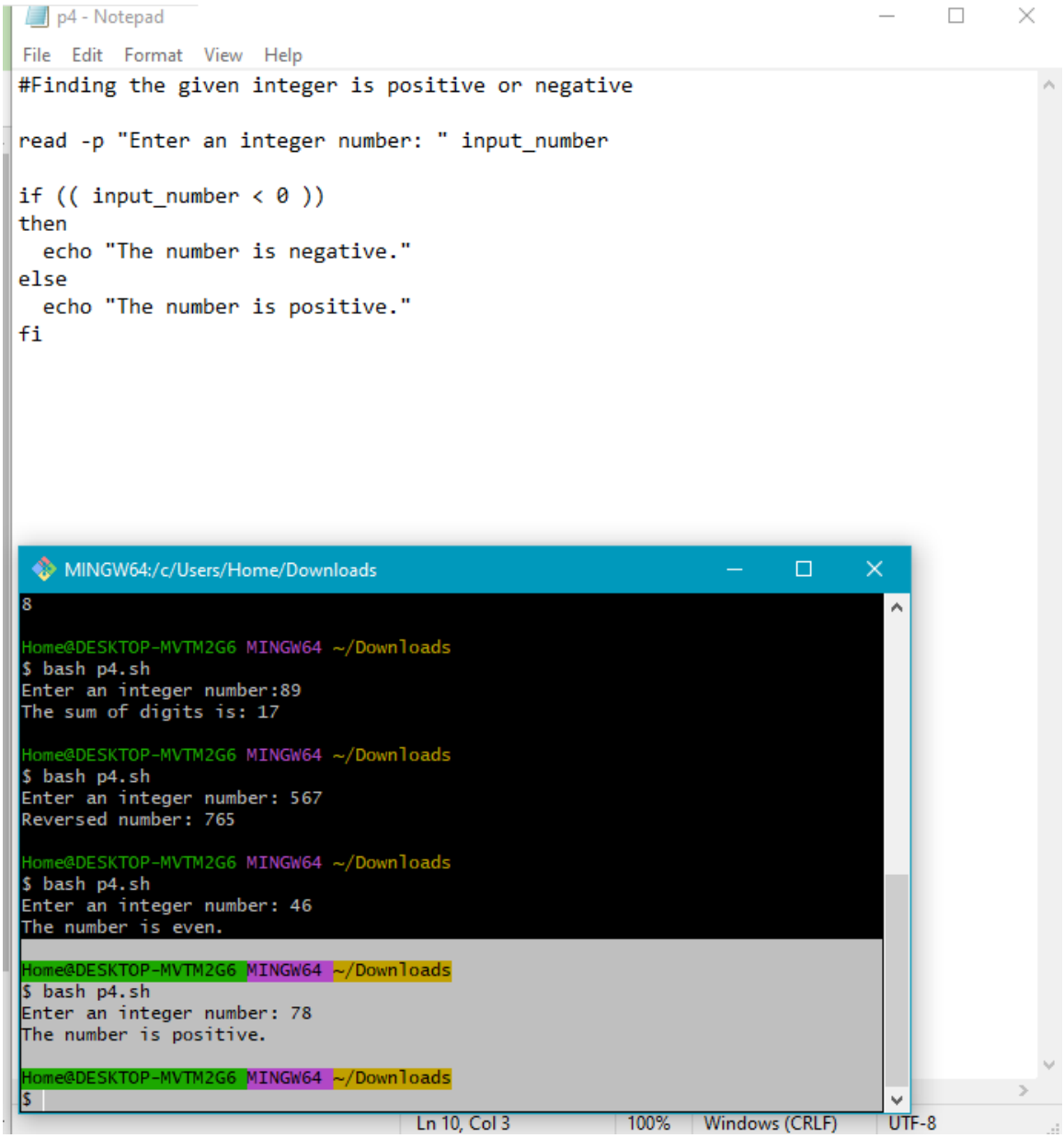
MINGW64:/c/Users/Home/Downloads

```
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer: 78
7
8

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number:89
The sum of digits is: 17

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 567
Reversed number: 765

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 46
The number is even.

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
```

Ln 8, Col 28 100% Windows (CRLF) UTF-8

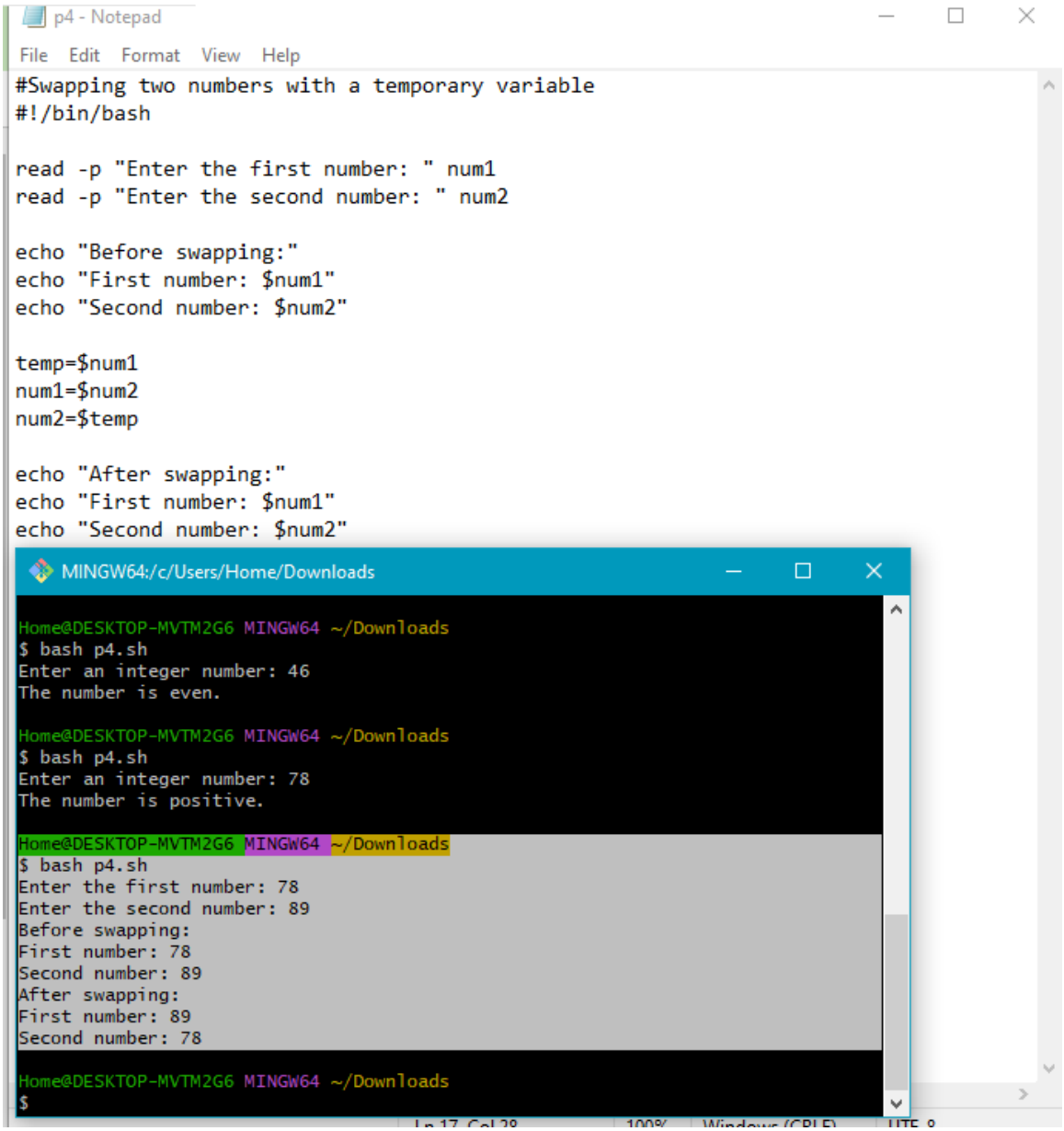
PROGRAM 21 :

CODE :


p4 - Notepad
File Edit Format View Help
#Finding the given integer is positive or negative
read -p "Enter an integer number: " input_number
if ((input_number < 0))
then
echo "The number is negative."
else
echo "The number is positive."
fi
MINGW64:/c/Users/Home/Downloads
8
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number:89
The sum of digits is: 17
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 567
Reversed number: 765
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 46
The number is even.
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 78
The number is positive.
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
Ln 10, Col 3
100%
Windows (CRLF)
UTF-8

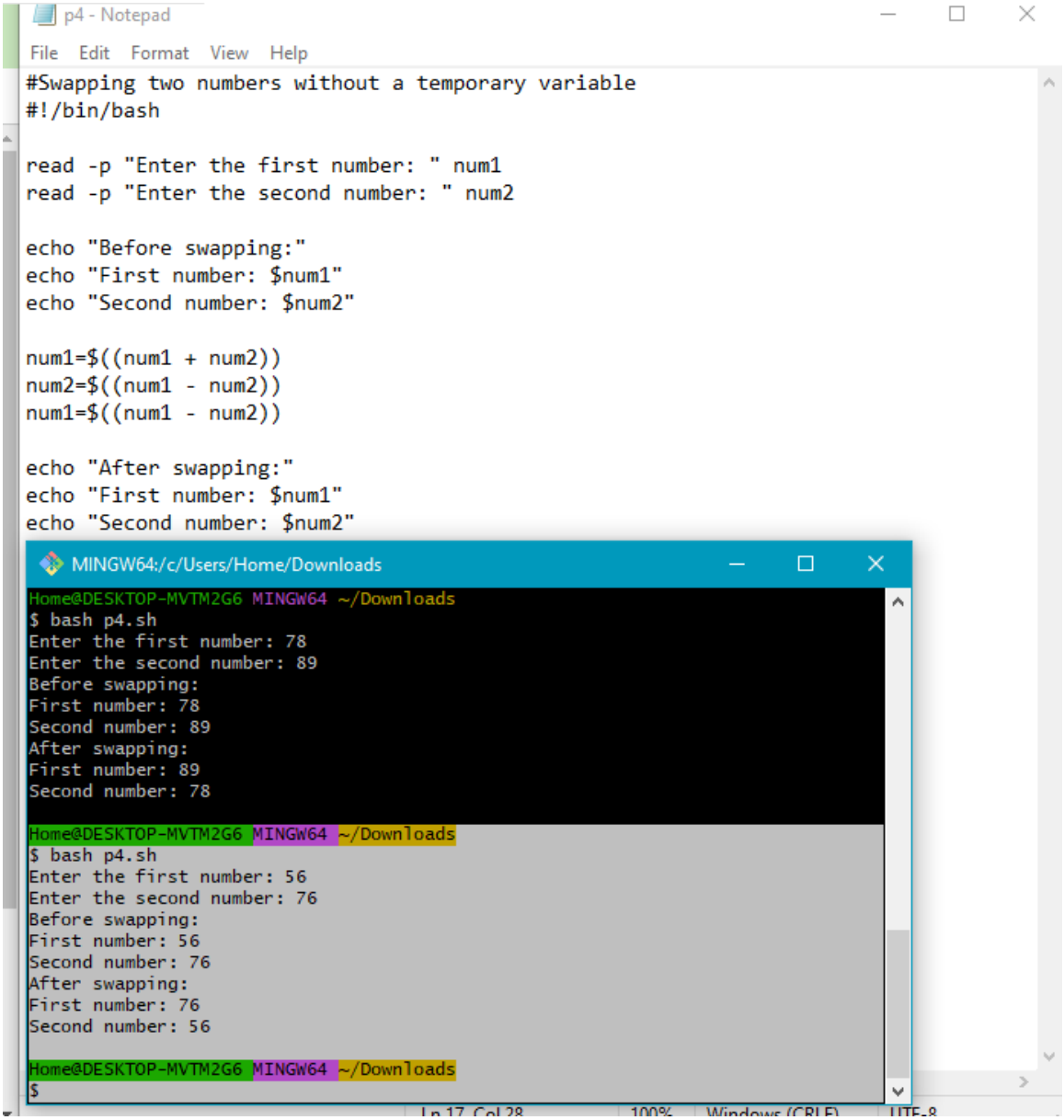
PROGRAM 22:

CODE:


p4 - Notepad
File Edit Format View Help
#Swapping two numbers with a temporary variable
#!/bin/bash
read -p "Enter the first number: " num1
read -p "Enter the second number: " num2
echo "Before swapping:"
echo "First number: $num1"
echo "Second number: $num2"
temp=$num1
num1=$num2
num2=$temp
echo "After swapping:"
echo "First number: $num1"
echo "Second number: $num2"
MINGW64:/c/Users/Home/Downloads
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 46
The number is even.
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter an integer number: 78
The number is positive.
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter the first number: 78
Enter the second number: 89
Before swapping:
First number: 78
Second number: 89
After swapping:
First number: 89
Second number: 78
Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$

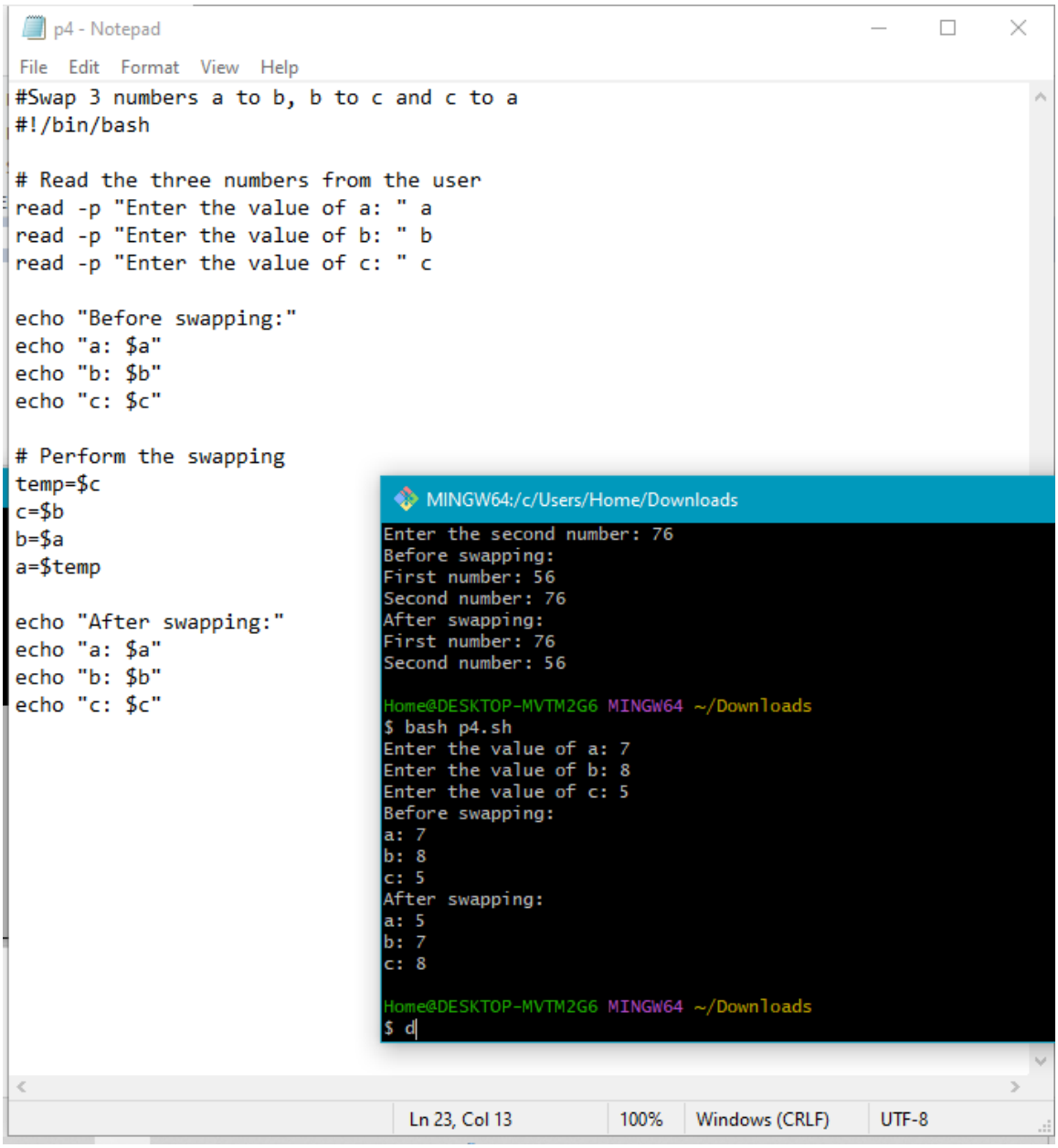
PROGRAM 23:

CODE :



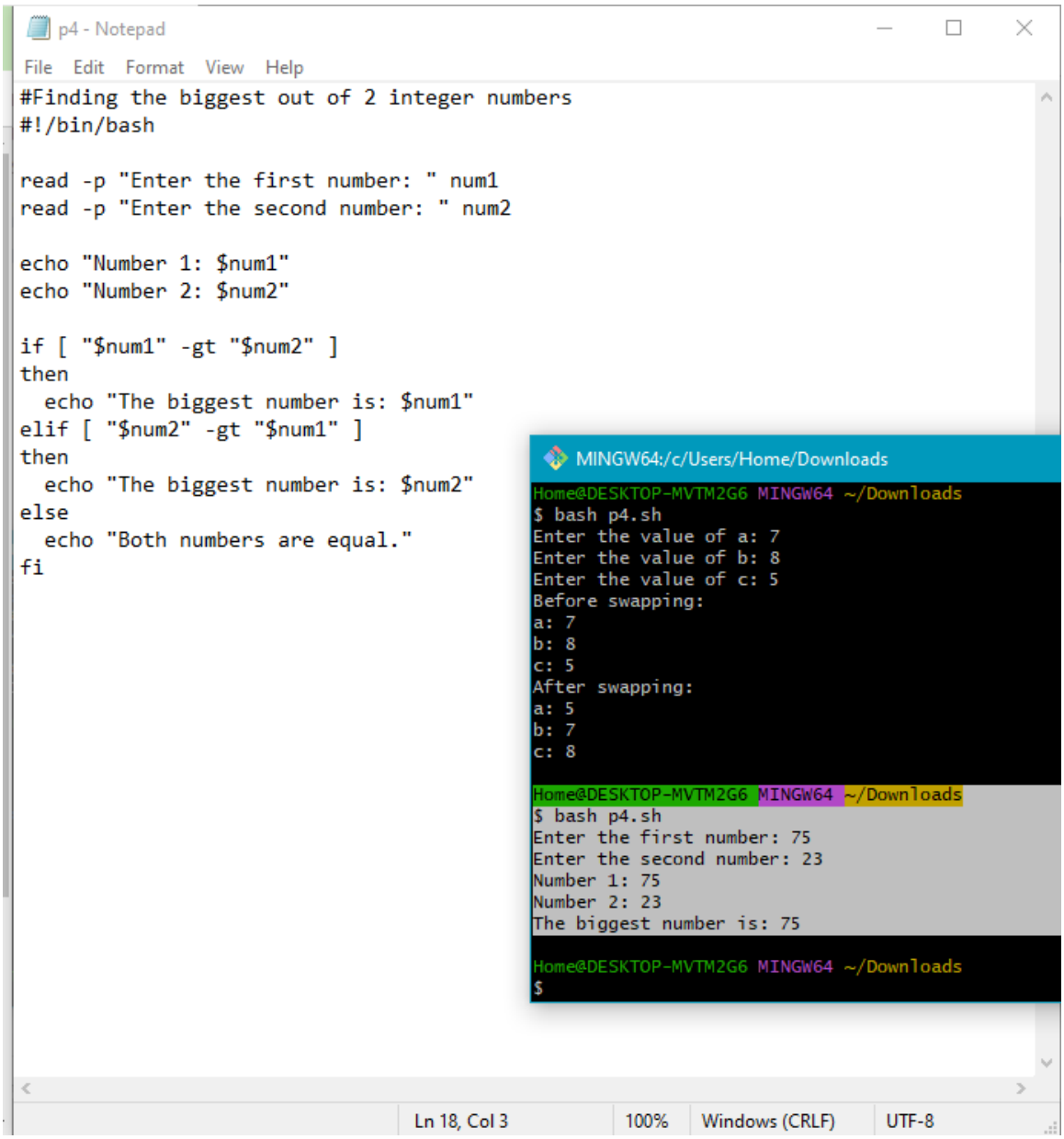
PROGRAM 24 :

CODE :



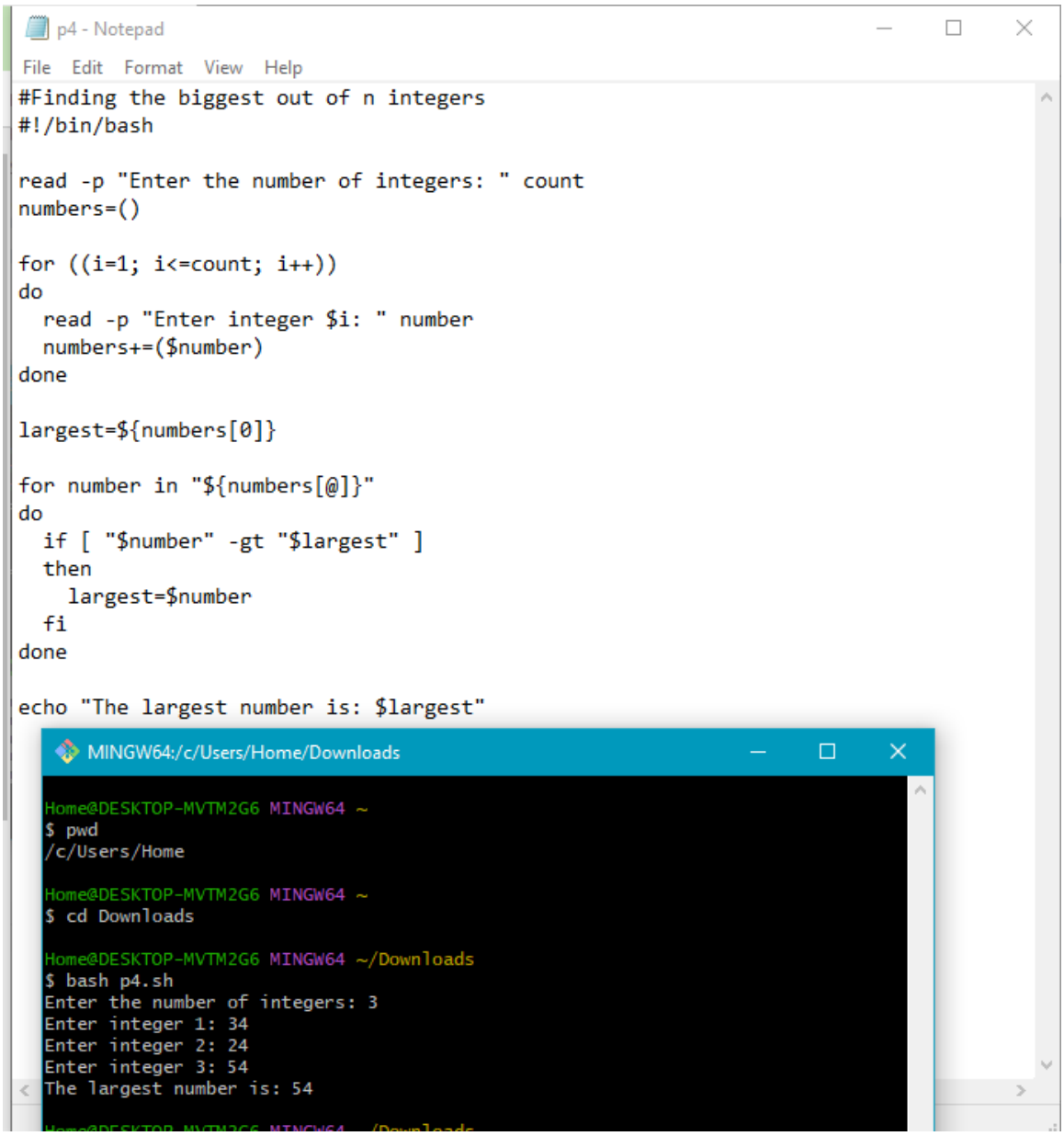
PROGRAM 25:

CODE :



PROGRAM 26 :

CODE:



PROGRAM 30:

CODE:

p4 - Notepad

File Edit Format View Help

```
#Linear Search
#!/bin/bash

# Function to perform linear search
linear_search() {
  local search_key=$1
  # Convert function arguments into an array
  local array=("$@")
  local found=false
  local index=0

  for element in "${array[@]}"
  do
    if [ "$element" -eq "$search_key" ]
    then
      found=true
      break
    fi
    index=$((index + 1))
  done

  if [ "$found" = true ]; then
    echo "Element $search_key found at index $index."
  else
    echo "Element $search_key not found in the array."
  fi
}

read -p "Enter the array elements (space-separated): " -a input_array

read -p "Enter the search key: " search_key

linear_search "$search_key" "${input_array[@]}"
```

MINGW64:/c/Users/Home/Downloads

```
Home@DESKTOP-MVTM2G6 MINGW64 ~
$ pwd
/c/Users/Home

Home@DESKTOP-MVTM2G6 MINGW64 ~
$ cd Downloads

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter the array elements (space-separated): 10
Enter the search key: 2
Element 2 found at index 0.

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$ bash p4.sh
Enter the array elements (space-separated): 20
Enter the search key: 15
Element 15 found at index 0.

Home@DESKTOP-MVTM2G6 MINGW64 ~/Downloads
$
```

Ln 33, Col 48 | 100% | Windows (CRLF) | UTF-8

Search | 32°C Mostly sunny | ENG | 10:22 AM