

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO CIENCIAS DE LA COMPUTACIÓN

SISTEMAS OPERATIVOS



## TAREAS

**NRC:** 14912

**Carrera:** Ingeniería de Software

**Nombre:** Gustavo Aguas - Sebastian Paucar

**Docente:** Ing. Fuertes Diaz Walter Marcelo Dr.

SANGOLQUI - ECUADOR

2024

# Capítulo 1

## Tareas

### 1.1. Tarea 1

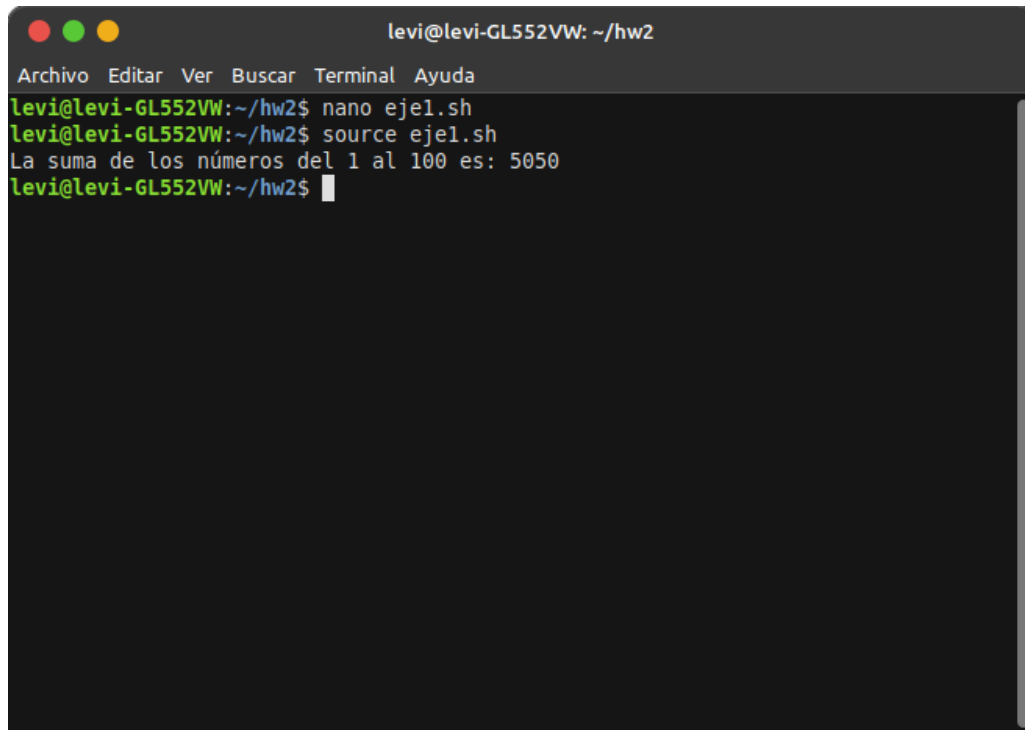
#### 1.1.1. 10 Programas que automaticen procesos

Realizar 10 ejercicios

#### 1.1.2. Ejercicio 1

```
#!/bin/bash
# Program Ejercicio 1
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa suma los n meros del 1 al
    100 utilizando un ciclo for.

sum=0
for i in {1..100}
do
    sum=$((sum + i))
done
echo "La suma de los n meros del 1 al 100 es: $sum"
# I love Linux
```

A terminal window titled 'levi@levi-GL552VW: ~/hw2' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
levi@levi-GL552VW:~/hw2$ nano ejel.sh
levi@levi-GL552VW:~/hw2$ source ejel.sh
La suma de los números del 1 al 100 es: 5050
levi@levi-GL552VW:~/hw2$
```

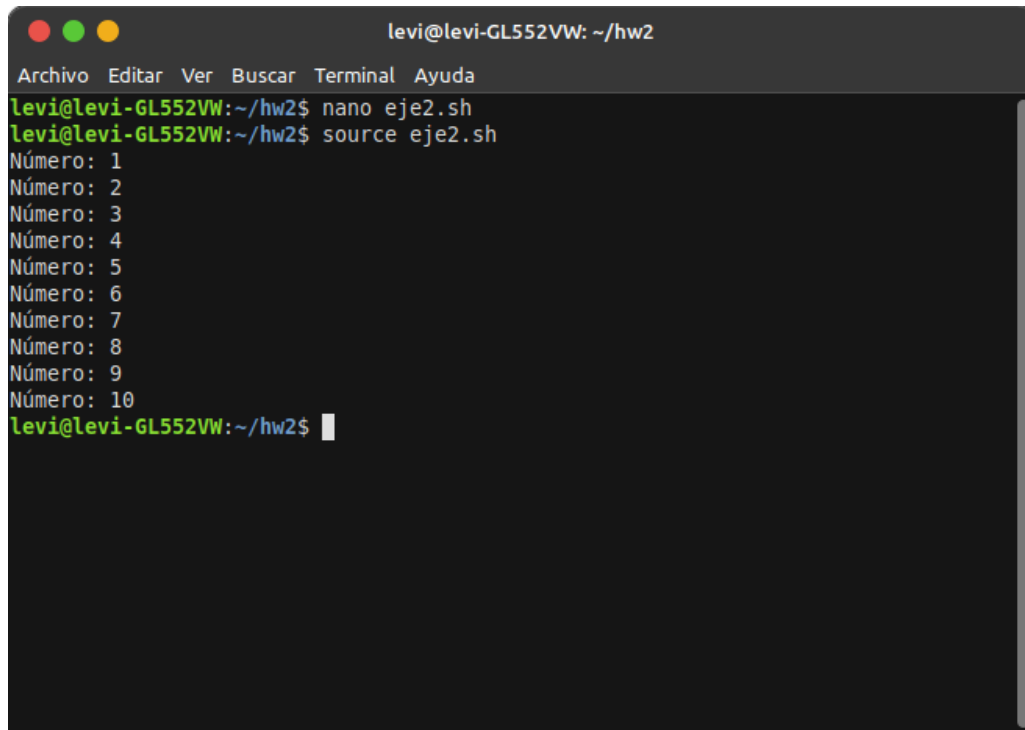
Figura 1.1: compilación del ejercicio 1

### 1.1.3. Ejercicio 2

```
#!/bin/bash
# Program Ejercicio 2
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa cuenta del 1 al 10
# utilizando un ciclo while.

count=1

while [ $count -le 10 ]
do
    echo "N mero: $count"
    count=$((count + 1))
done
# I love Linux
```



```
levi@levi-GL552VW: ~/hw2
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
levi@levi-GL552VW:~/hw2$ nano eje2.sh
levi@levi-GL552VW:~/hw2$ source eje2.sh
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
Número: 6
Número: 7
Número: 8
Número: 9
Número: 10
levi@levi-GL552VW:~/hw2$
```

Figura 1.2: compilación del ejercicio 2

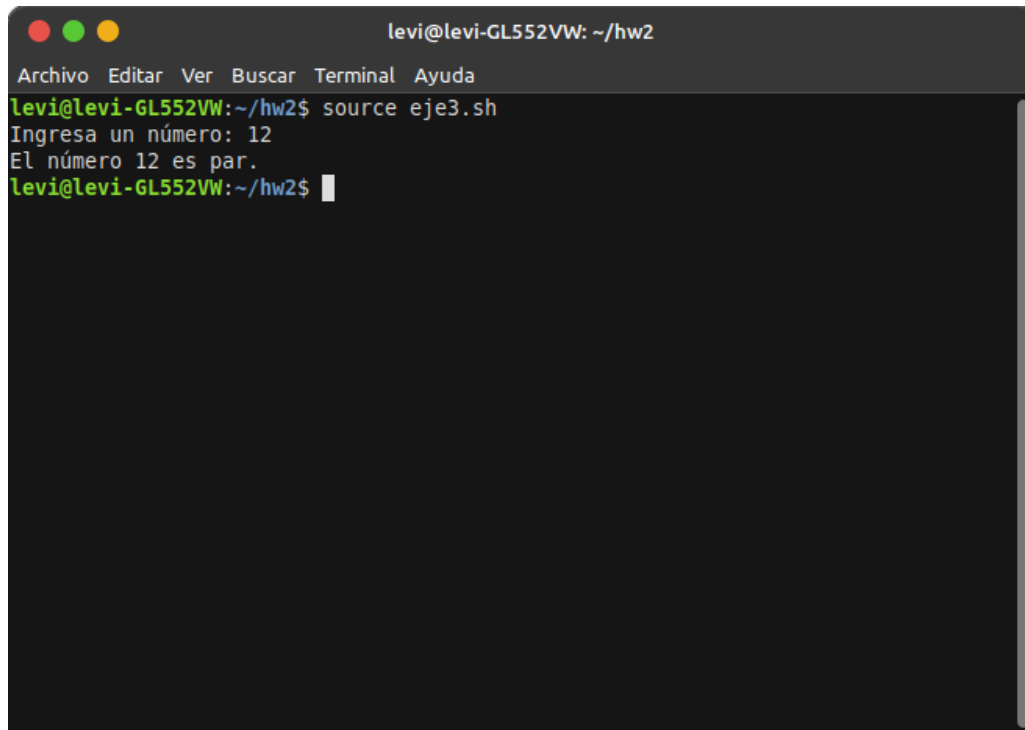
#### 1.1.4. Ejercicio 3

```
#!/bin/bash
# Program Ejercicio 3
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa verifica si un n mero
# ingresado es par o impar.

read -p "Ingresa un n mero: " num

if [ $((num % 2)) -eq 0 ]; then
    echo "El n mero $num es par."
else
    echo "El n mero $num es impar."
fi

#I love linux
```

A terminal window titled 'levi@levi-GL552VW: ~/hw2' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the command 'source eje3.sh' being executed. The script prompts 'Ingresa un número: 12' and outputs 'El número 12 es par.' before returning to the shell prompt.

```
levi@levi-GL552VW: ~/hw2
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~/hw2$ source eje3.sh
Ingresa un número: 12
El número 12 es par.
levi@levi-GL552VW:~/hw2$
```

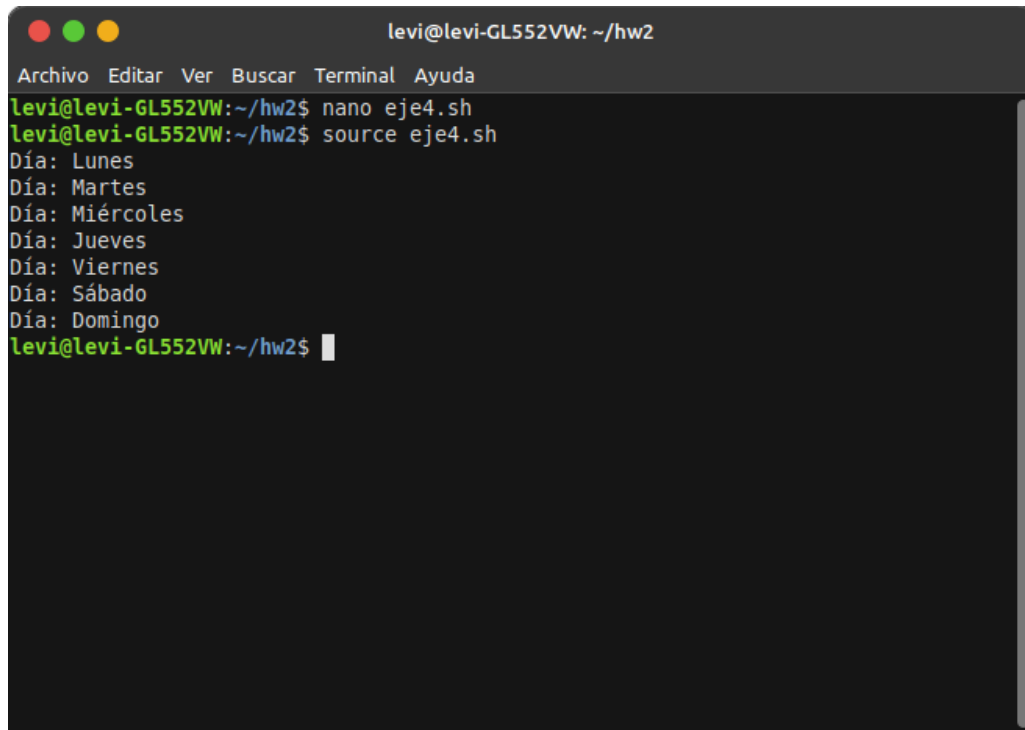
Figura 1.3: compilación del ejercicio 3

#### 1.1.5. Ejercicio 4

```
#!/bin/bash
# Program Ejercicio 4
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa muestra los nombres de los
#   días de la semana utilizando un arreglo.

dias=("Lunes" "Martes" "Miércoles" "Jueves" "Viernes" "Sábado" "
Domingo")

for dia in "${dias[@]}"
do
    echo "Día: $dia"
done
#I love linux
```

A terminal window titled 'levi@levi-GL552VW: ~/hw2' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
levi@levi-GL552VW:~/hw2$ nano eje4.sh
levi@levi-GL552VW:~/hw2$ source eje4.sh
Día: Lunes
Día: Martes
Día: Miércoles
Día: Jueves
Día: Viernes
Día: Sábado
Día: Domingo
levi@levi-GL552VW:~/hw2$
```

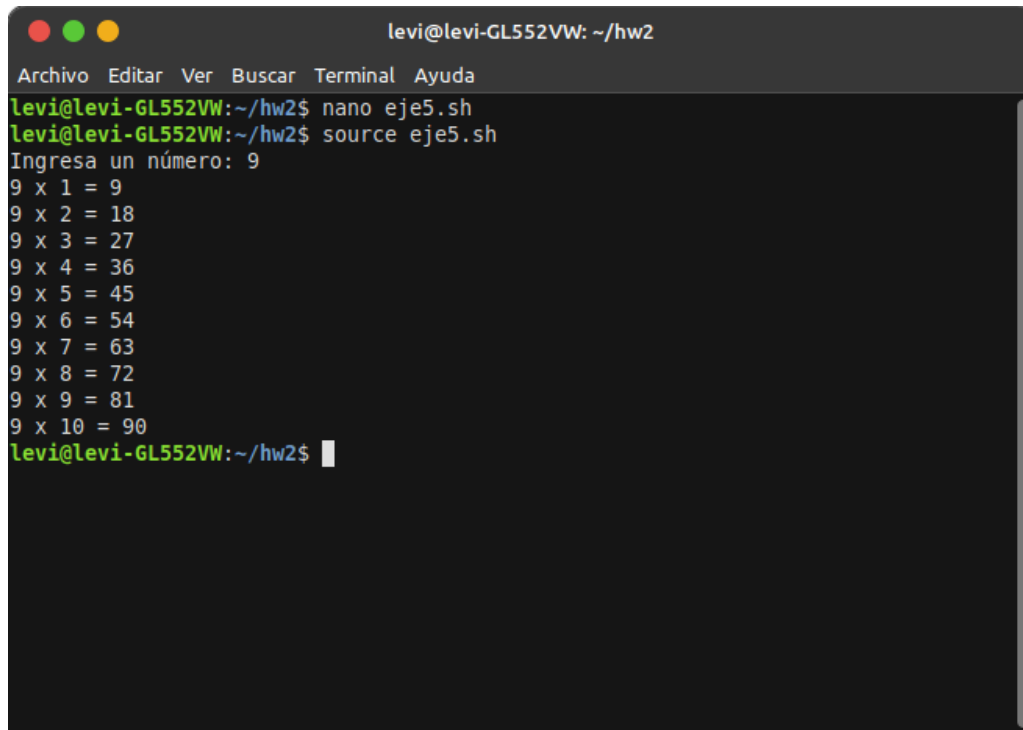
Figura 1.4: compilación del ejercicio 4

### 1.1.6. Ejercicio 5

```
#!/bin/bash
# Program Ejercicio 5
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa imprime la tabla de
# multiplicar del n mero ingresado.

read -p "Ingresa un n mero: " num

for i in {1..10}
do
    echo "$num x $i = $((num * i))"
done
#I love linux
```

A screenshot of a terminal window titled 'levi@levi-GL552VW: ~/hw2'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
levi@levi-GL552VW:~/hw2$ nano eje5.sh
levi@levi-GL552VW:~/hw2$ source eje5.sh
Ingresa un número: 9
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
levi@levi-GL552VW:~/hw2$
```

Figura 1.5: compilación del ejercicio 5

### 1.1.7. Ejercicio 6

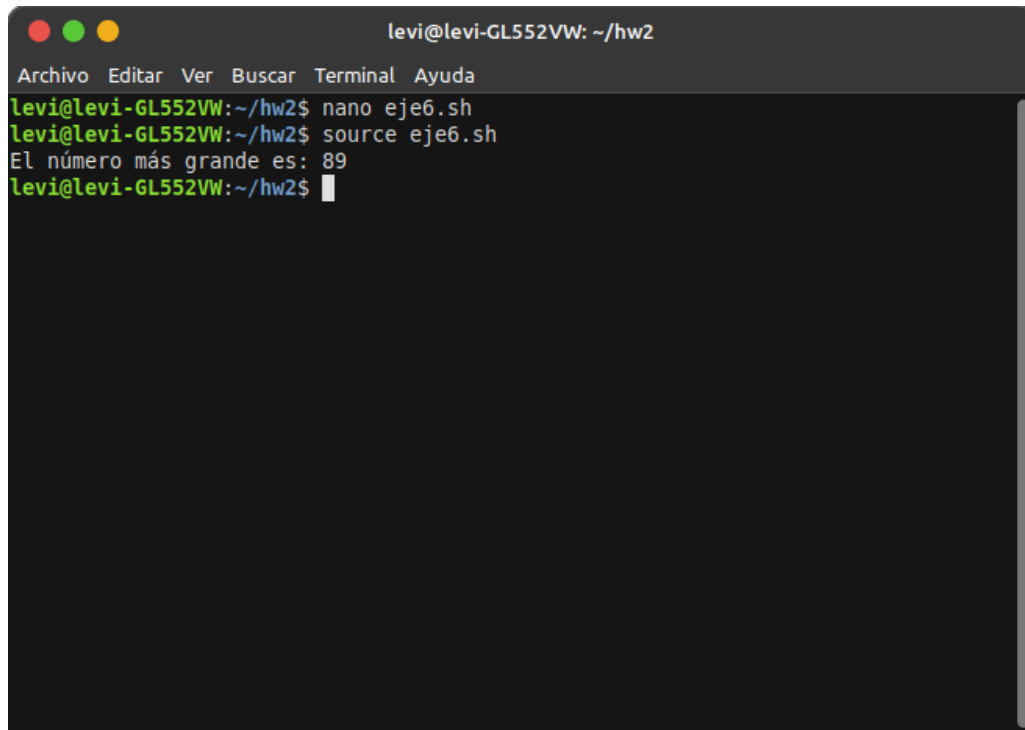
```
#!/bin/bash
# Program Ejercicio 6
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa encuentra el n mero m s
# grande en un arreglo.

numeros=(23 45 67 89 12 34 56)

max=${numeros[0]}

for num in "${numeros[@]}"
do
    if [ $num -gt $max ]; then
        max=$num
    fi
done

echo "El n mero m s grande es: $max"
#I love linux
```

A terminal window titled 'levi@levi-GL552VW: ~/hw2' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
levi@levi-GL552VW:~/hw2$ nano eje6.sh
levi@levi-GL552VW:~/hw2$ source eje6.sh
El número más grande es: 89
levi@levi-GL552VW:~/hw2$
```

Figura 1.6: compilación del ejercicio 6

### 1.1.8. Ejercicio 7

```
#!/bin/bash
# Program Ejercicio 7
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa cuenta el n mero de vocales
# en una cadena ingresada.

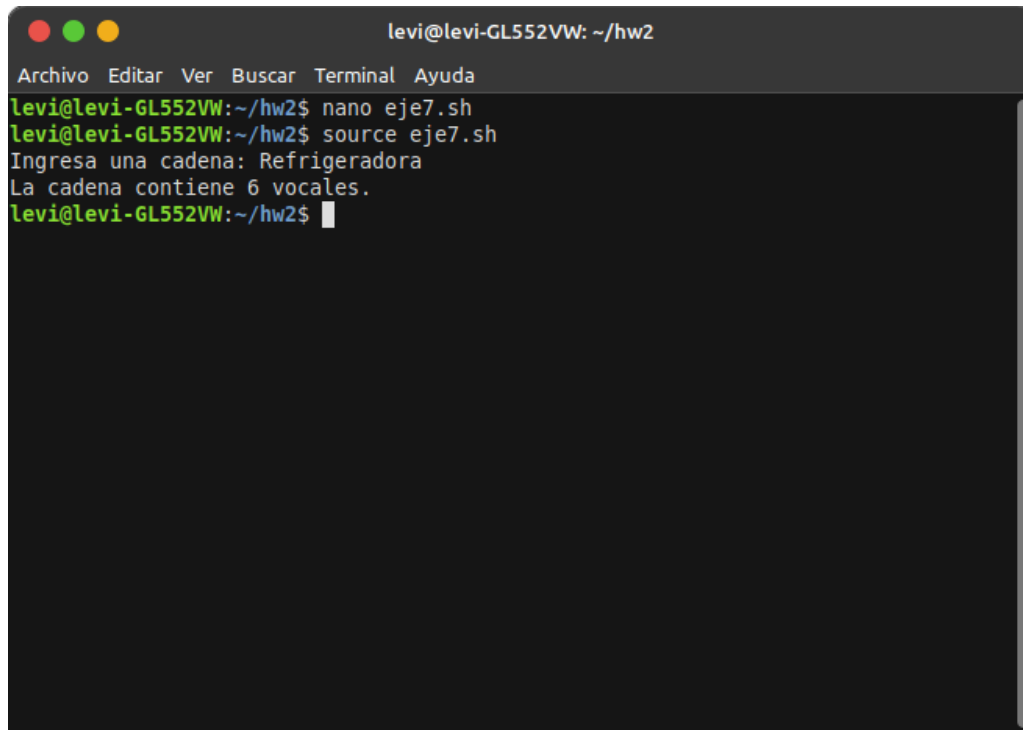
read -p "Ingresa una cadena: " cadena

vocales=0

for (( i=0; i<${#cadena}; i++ ))
do
    char=${cadena:$i:1}
    if [[ "$char" =~ [aeiouAEIOU] ]]; then
        vocales=$((vocales + 1))
    fi
done

echo "La cadena contiene $vocales vocales."
#I love linux
```





```
levi@levi-GL552VW: ~/hw2
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~/hw2$ nano eje7.sh
levi@levi-GL552VW:~/hw2$ source eje7.sh
Ingresa una cadena: Refrigeradora
La cadena contiene 6 vocales.
levi@levi-GL552VW:~/hw2$
```

Figura 1.7: compilación del ejercicio 7

### 1.1.9. Ejercicio 8

```
#!/bin/bash
# Program Ejercicio 8
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa verifica si un n mero
# ingresado es primo.

read -p "Ingresa un n mero: " num
es_primo=1

if [ $num -le 1 ]; then
    es_primo=0
else
    for ((i=2; i<=num/2; i++))
    do
        if [ $((num % i)) -eq 0 ]; then
            es_primo=0
            break
        fi
    done
fi

if [ $es_primo -eq 1 ]; then
    echo "El n mero $num es primo."
```

```
else
    echo "El n mero $num no es primo."
fi
#I love linux
```

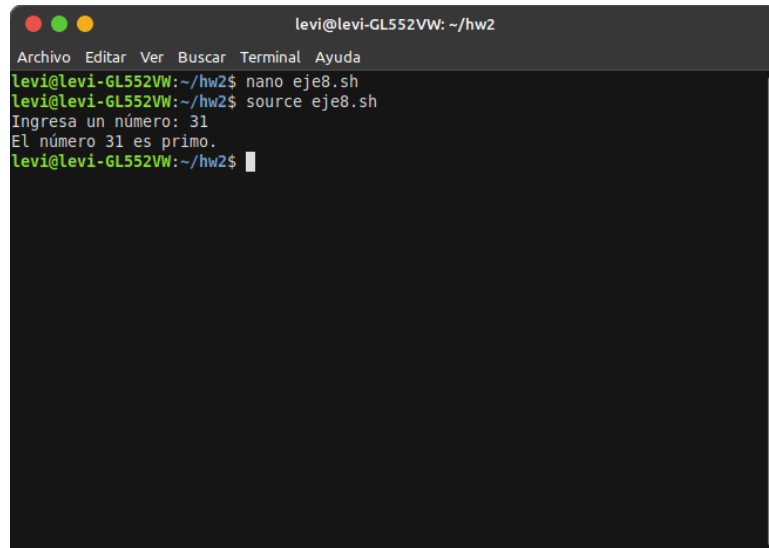


Figura 1.8: compilación del ejercicio 8

**1.1.10. Ejercicio 9**

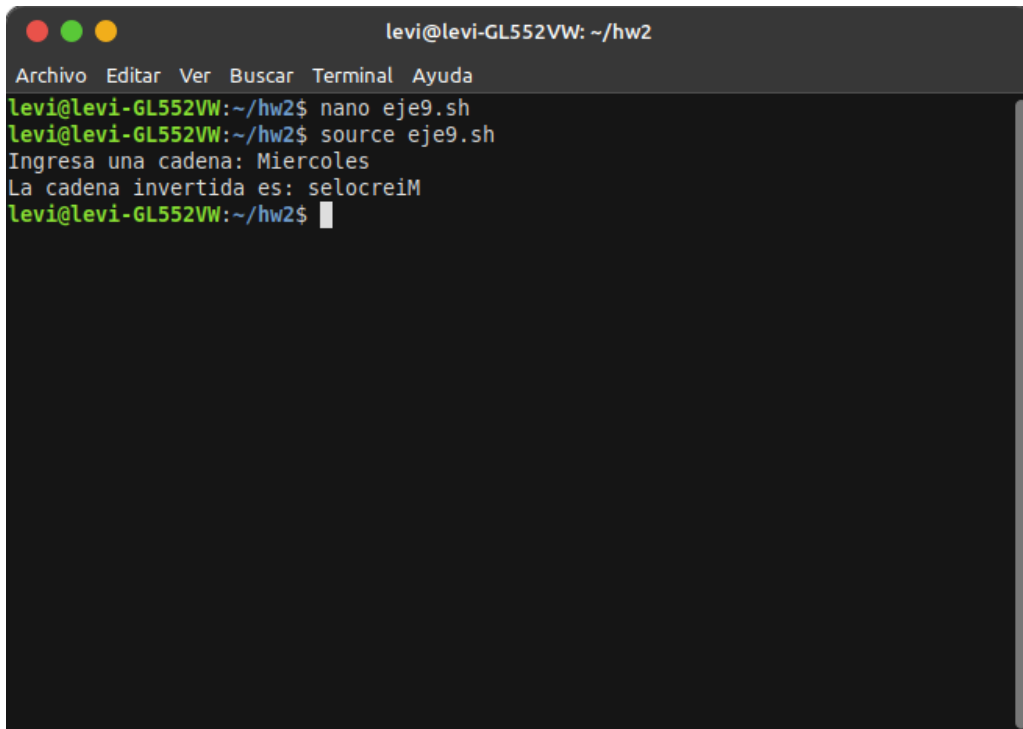
```
#!/bin/bash
# Program Ejercicio 9
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa invierte una cadena
# ingresada.

read -p "Ingresa una cadena: " cadena

longitud=${#cadena}

for (( i=$longitud-1; i>=0; i-- ))
do
    invertida="$invertida${cadena:$i:1}"
done

echo "La cadena invertida es: $invertida"
#I love linux
```



```
levi@levi-GL552VW: ~/hw2
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~/hw2$ nano eje9.sh
levi@levi-GL552VW:~/hw2$ source eje9.sh
Ingresa una cadena: Miercoles
La cadena invertida es: selocreiM
levi@levi-GL552VW:~/hw2$
```

Figura 1.9: compilación del ejercicio 9

**1.1.11. Ejercicio 10**

```
#!/bin/bash
# Program Ejercicio 10
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este programa calcula el factorial de un
  n mero ingresado.

read -p "Ingresa un n mero: " num
factorial=1

for (( i=1; i<=num; i++ ))
do
    factorial=$((factorial * i))
done

echo "El factorial de $num es: $factorial"
```

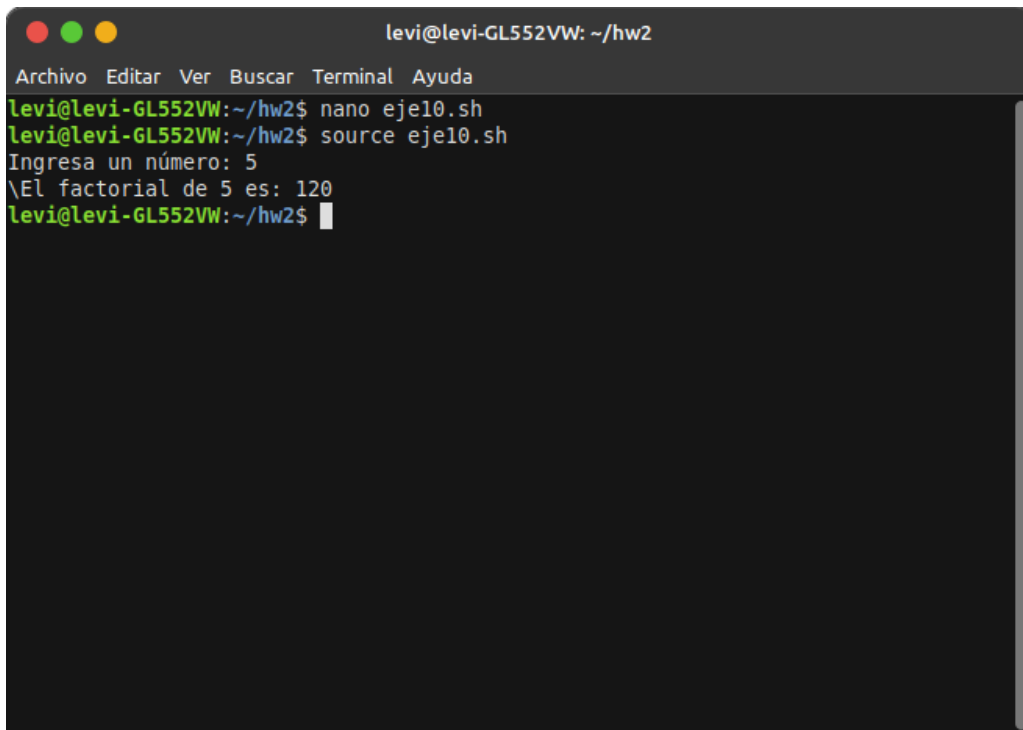
A screenshot of a terminal window titled "levi@levi-GL552VW: ~/hw2". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal shows the following commands and output:  
levi@levi-GL552VW:~/hw2\$ nano eje10.sh  
levi@levi-GL552VW:~/hw2\$ source eje10.sh  
Ingresa un número: 5  
\El factorial de 5 es: 120  
levi@levi-GL552VW:~/hw2\$

Figura 1.10: compilación del ejercicio 10

## 1.2. Tarea 2

### 1.2.1. Ejercicios de programación en Shell con while, if

Realizar 15 ejercicios

#### 1.2.2. Ejercicio 1

El menor de dos numeros

```
#!/bin/bash
#Program Ejercicio 1
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
clear
echo "Ingrese el 1er numero";
read num1;
echo "Ingrese el 2do numero";
read num2;
if [ $num1 -lt $num2 ]; then
echo "el menor numero es: $num1"
elif [ $num2 -lt $num1 ]; then
echo "El menor es: $num2";
else
echo "$num1, $num2 son iguales";
fi
#I love linux
```

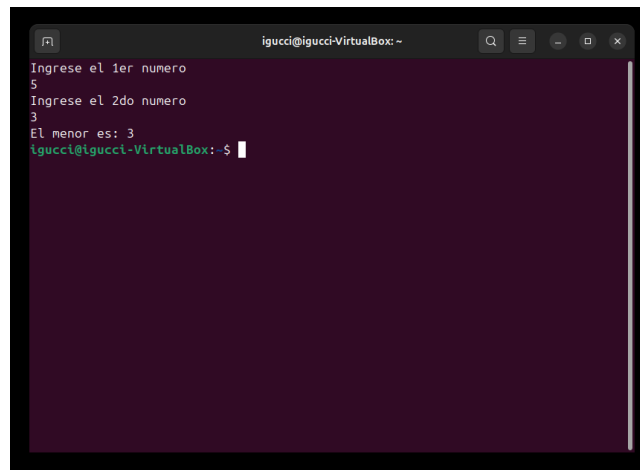


Figura 1.11: Ejercicio 1

### 1.2.3. Ejercicio 2

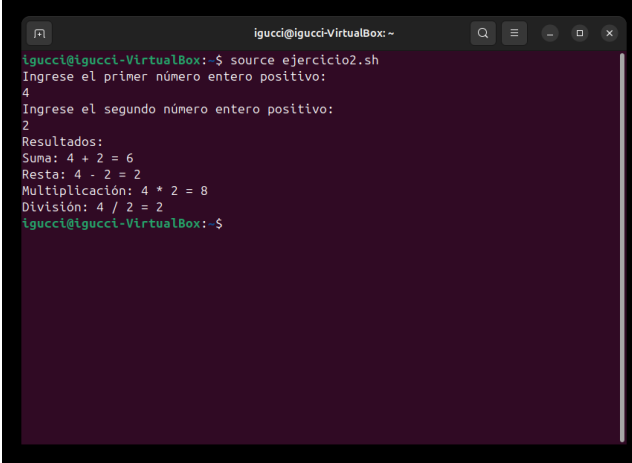
Operaciones basicas

```
#!/bin/bash
#Program Ejercicio 2
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "Ingrese el primer n mero entero positivo:"
read num1

echo "Ingrese el segundo n mero entero positivo:"
read num2

suma=$(expr $num1 + $num2)
resta=$(expr $num1 - $num2)
multiplicacion=$(expr $num1 \* $num2)
division=$(expr $num1 / $num2)

echo "Resultados:"
echo "Suma: $num1 + $num2 = $suma"
echo "Resta: $num1 - $num2 = $resta"
echo "Multiplicaci n: $num1 * $num2 = $multiplicacion"
echo "Divisi n: $num1 / $num2 = $division"
```



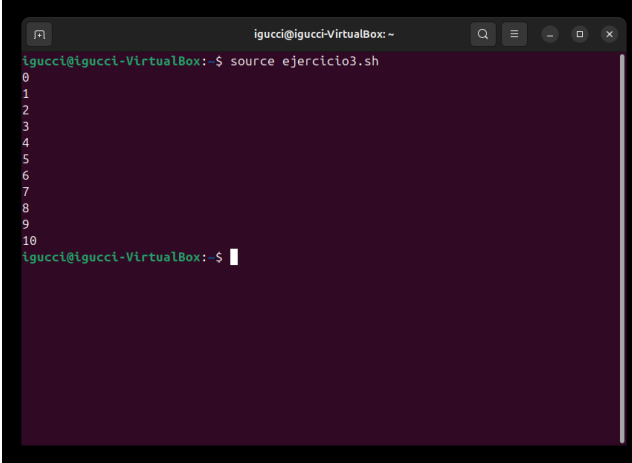
```
igucci@igucci-VirtualBox: ~
igucci@igucci-VirtualBox:~$ source ejercicio2.sh
Ingrese el primer número entero positivo:
4
Ingrese el segundo número entero positivo:
2
Resultados:
Suma: 4 + 2 = 6
Resta: 4 - 2 = 2
Multiplicación: 4 * 2 = 8
División: 4 / 2 = 2
igucci@igucci-VirtualBox:~$
```

Figura 1.12: Ejercicio 2

### 1.2.4. Ejercicio 3

Secuencia hasta el 10

```
#!/bin/bash
#Program Ejercicio 3
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
cont=0
while [ $cont -lt 11 ]
do
echo $cont
cont=$((cont + 1))
done
# I LOVE LINUX
```

A screenshot of a terminal window titled 'igucci@igucci-VirtualBox: ~'. The terminal shows the command 'source ejercicio3.sh' being executed. The output of the script is a list of numbers from 0 to 10, each on a new line. The prompt 'igucci@igucci-VirtualBox:~\$' is visible at the bottom of the terminal window.

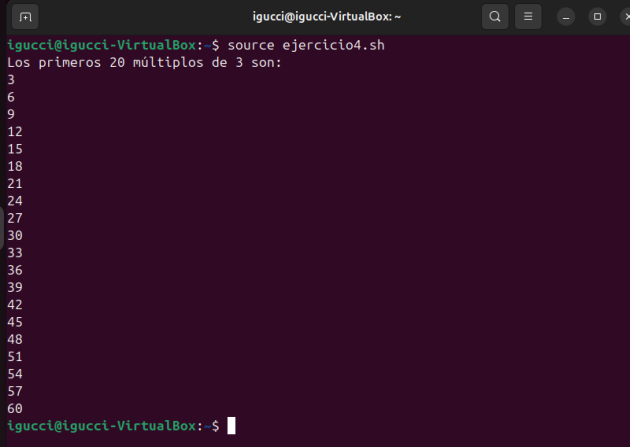
```
igucci@igucci-VirtualBox:~$ source ejercicio3.sh
0
1
2
3
4
5
6
7
8
9
10
igucci@igucci-VirtualBox:~$
```

Figura 1.13: Ejercicio 3

### 1.2.5. Ejercicio 4

Primeros 20 multiples de 3

```
#!/bin/bash
#Program Ejercicio 4
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
# Imprimir los primeros 20 mltiplos de 3
echo "Los primeros 20 mltiplos de 3 son:"
for (( i=1; i<=20; i++ )); do
    multiplo=$((i * 3))
    echo $multiplo
done
```

A screenshot of a terminal window titled 'igucci@igucci-VirtualBox: ~'. The prompt is 'igucci@igucci-VirtualBox:~\$'. The user has entered 'source ejercicio4.sh'. The output of the script is displayed: 'Los primeros 20 múltiplos de 3 son:' followed by a list of numbers from 3 to 60 in increments of 3. The prompt 'igucci@igucci-VirtualBox:~\$' is visible at the bottom with a cursor.

```
igucci@igucci-VirtualBox:~$ source ejercicio4.sh
Los primeros 20 múltiplos de 3 son:
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
igucci@igucci-VirtualBox:~$
```

Figura 1.14: Ejercicio 4

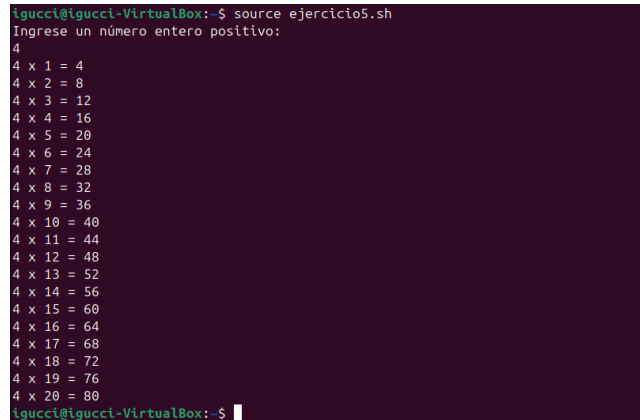


### 1.2.6. Ejercicio 5

Tabla de multiplicar del num 4 hasta el 20

```
#!/bin/bash
#Program Ejercicio 5
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
# Leer un n mero entero positivo
echo "Ingrese un n mero entero positivo:"
read num

# Calcular y mostrar la tabla de multiplicar hasta el 20
for ((i = 1; i <= 20; i++)); do
    resultado=$((num * i))
    echo "$num x $i = $resultado"
done
```



```
igucci@igucci-VirtualBox: $ source ejercicio5.sh
Ingrese un número entero positivo:
4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
4 x 11 = 44
4 x 12 = 48
4 x 13 = 52
4 x 14 = 56
4 x 15 = 60
4 x 16 = 64
4 x 17 = 68
4 x 18 = 72
4 x 19 = 76
4 x 20 = 80
igucci@igucci-VirtualBox: $
```

Figura 1.15: Ejercicio 5

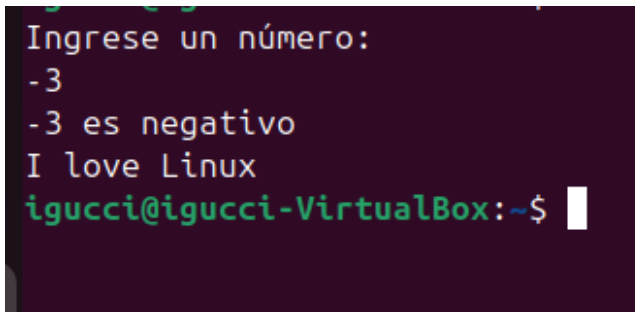
### 1.2.7. Ejercicio 6

Verificar si un número es positivo, negativo o cero

```
#!/bin/bash
#Program Ejercicio 6
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "Ingrese un n mero:"
read num

if (( num > 0 )); then
    echo "$num es positivo"
elif (( num < 0 )); then
    echo "$num es negativo"
else
    echo "$num es cero"
fi

echo "I love Linux"
```



```
Ingrese un número:
-3
-3 es negativo
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.16: Ejercicio 6

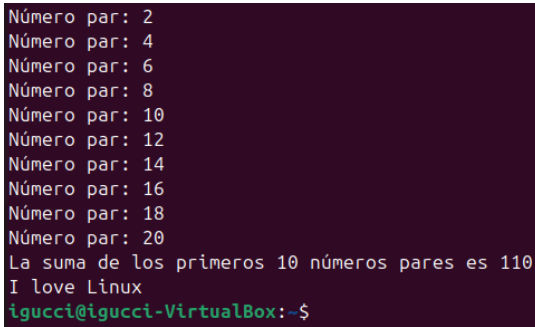
### 1.2.8. Ejercicio 7

Imprimir los primeros 10 números pares y su suma

```
#!/bin/bash
#Program Ejercicio 7
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
suma=0
for (( i = 1; i <= 10; i++ )); do
    num=$(( i * 2 ))
    echo "Número par: $num"
    suma=$(( suma + num ))
done

echo "La suma de los primeros 10 números pares es $suma"

echo "I love Linux"
```



```
Número par: 2
Número par: 4
Número par: 6
Número par: 8
Número par: 10
Número par: 12
Número par: 14
Número par: 16
Número par: 18
Número par: 20
La suma de los primeros 10 números pares es 110
I love Linux
igucci@igucci-VirtualBox:~$
```

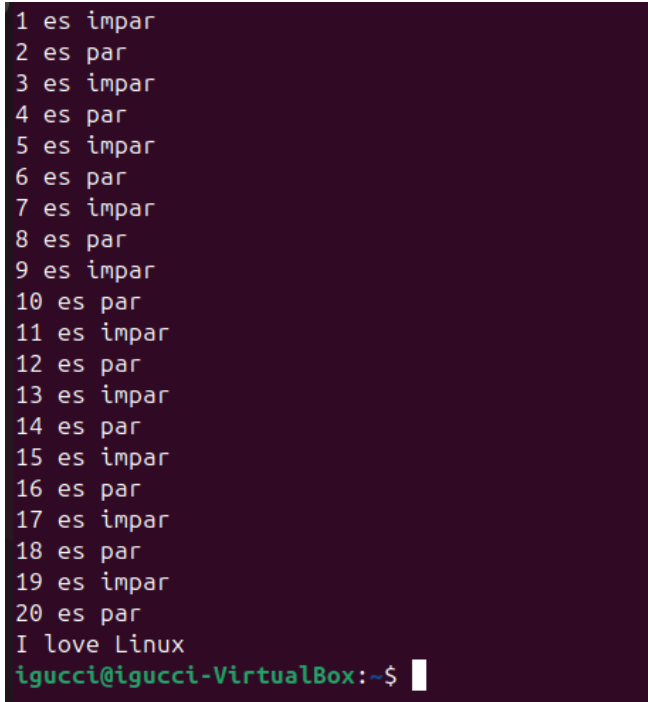
Figura 1.17: Ejercicio 7

### 1.2.9. Ejercicio 8

Contar del 1 al 20 y mostrar si cada número es par o impar

```
#!/bin/bash
#Program Ejercicio 8
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
for (( i = 1; i <= 20; i++ )); do
    if (( i % 2 == 0 )); then
        echo "$i es par"
    else
        echo "$i es impar"
    fi
done

echo "I love Linux"
```



```
1 es impar
2 es par
3 es impar
4 es par
5 es impar
6 es par
7 es impar
8 es par
9 es impar
10 es par
11 es impar
12 es par
13 es impar
14 es par
15 es impar
16 es par
17 es impar
18 es par
19 es impar
20 es par
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.18: Ejercicio 8

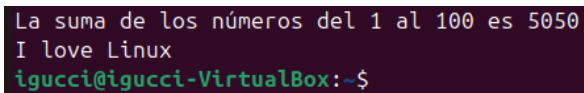
**1.2.10. Ejercicio 9**

Calcular la suma de los números del 1 al 100 y mostrar el resultado

```
#!/bin/bash
#Program Ejercicio 9
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
suma=0
for (( i = 1; i <= 100; i++ )); do
    suma=$(( suma + i ))
done

echo "La suma de los n meros del 1 al 100 es $suma"

echo "I love Linux"
```



```
La suma de los números del 1 al 100 es 5050
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.19: Ejercicio 9

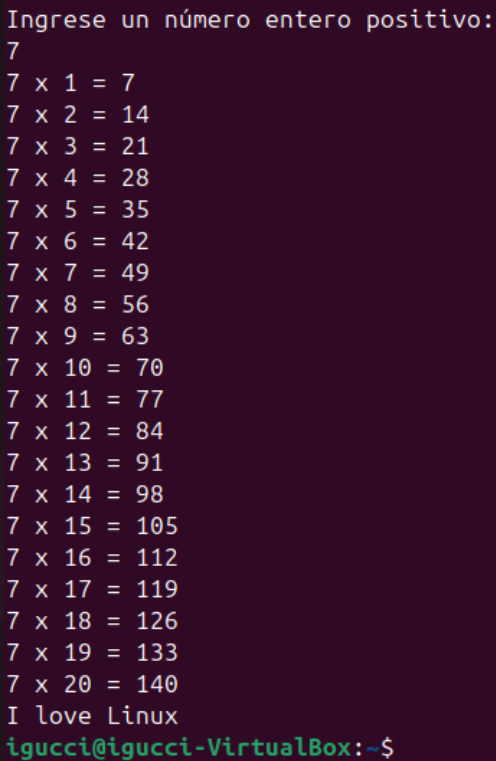
### 1.2.11. Ejercicio 10

Imprimir la tabla de multiplicar de un número hasta el 20

```
#!/bin/bash
#Program Ejercicio 10
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "Ingrese un n mero entero positivo:"
read num

for (( i = 1; i <= 20; i++ )); do
    resultado=$(( num * i ))
    echo "$num x $i = $resultado"
done

echo "I love Linux"
```



```
Ingrese un número entero positivo:
7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
7 x 11 = 77
7 x 12 = 84
7 x 13 = 91
7 x 14 = 98
7 x 15 = 105
7 x 16 = 112
7 x 17 = 119
7 x 18 = 126
7 x 19 = 133
7 x 20 = 140
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.20: Ejercicio 10

### 1.2.12. Ejercicio 11

Contar las letras de una cadena y mostrar cuántas son vocales y cuántas son consonantes

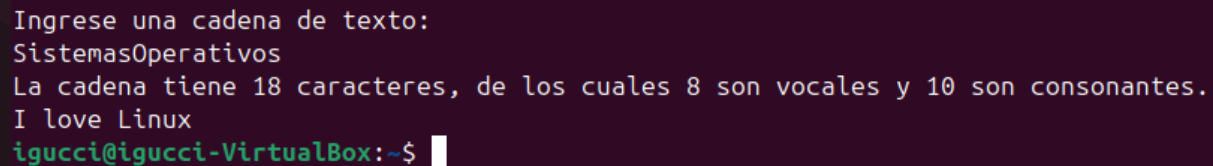
```
#!/bin/bash
#Program Ejercicio 11
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "Ingrese una cadena de texto:"
read cadena

longitud=${#cadena}
vocales=0
consonantes=0

for (( i = 0; i < longitud; i++ )); do
    letra=${cadena:$i:1}
    if [[ $letra =~ [aeiouAEIOU] ]]; then
        vocales=$(( vocales + 1 ))
    elif [[ $letra =~ [bcd fghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ] ]];
        then
            consonantes=$(( consonantes + 1 ))
        fi
    done

echo "La cadena tiene $longitud caracteres, de los cuales $vocales son
    vocales y $consonantes son consonantes."

echo "I love Linux"
```



```
Ingrese una cadena de texto:
SistemasOperativos
La cadena tiene 18 caracteres, de los cuales 8 son vocales y 10 son consonantes.
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.21: Ejercicio 11

**1.2.13. Ejercicio 12**

Comprobar si un número es múltiplo de 3 o 5 y mostrar todos los múltiplos de 3 o 5 hasta 50

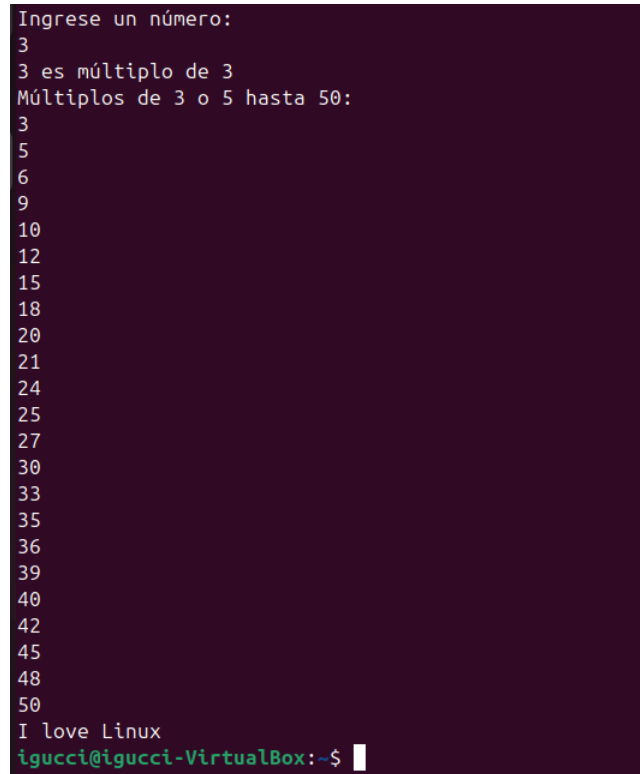
```
#!/bin/bash
#Program Ejercicio 12
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "Ingrese un n mero:"
read num

if (( num % 3 == 0 )); then
    echo "$num es m ltiplo de 3"
elif (( num % 5 == 0 )); then
    echo "$num es m ltiplo de 5"
else
    echo "$num no es m ltiplo de 3 ni de 5"
fi

echo "M ltiplos de 3 o 5 hasta 50:"
for (( i = 1; i <= 50; i++ )); do
    if (( i % 3 == 0 || i % 5 == 0 )); then
        echo $i
    fi
done

echo "I love Linux"
```





```
Ingrese un número:
3
3 es múltiplo de 3
Múltiplos de 3 o 5 hasta 50:
3
5
6
9
10
12
15
18
20
21
24
25
27
30
33
35
36
39
40
42
45
48
50
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.22: Ejercicio 12

### 1.2.14. Ejercicio 13

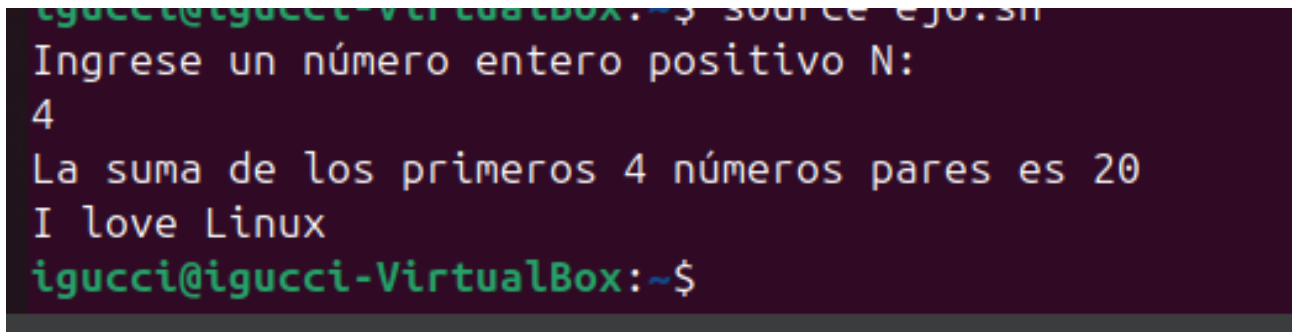
Sumar los primeros N números pares

```
#!/bin/bash
#Program Ejercicio 13
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "Ingrese un n mero entero positivo N:"
read N

suma=0
for (( i = 1; i <= N; i++ )); do
    num=$(( i * 2 ))
    suma=$(( suma + num ))
done

echo "La suma de los primeros $N n meros pares es $suma"

echo "I love Linux"
```



```
igucci@igucci-VirtualBox:~$ source ejer13.sh
Ingrese un número entero positivo N:
4
La suma de los primeros 4 números pares es 20
I love Linux
igucci@igucci-VirtualBox:~$
```

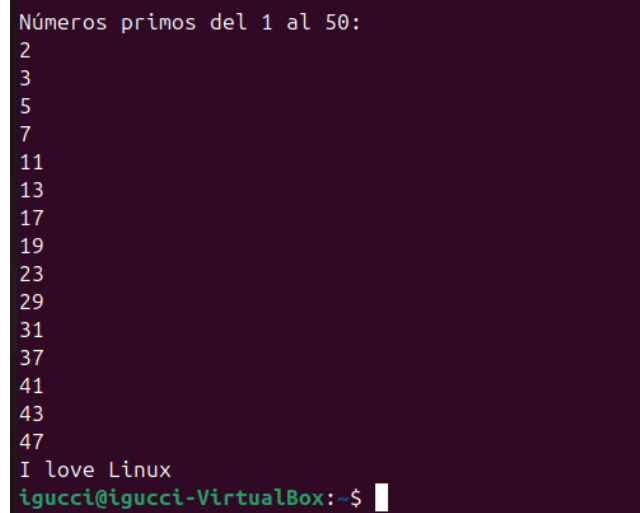
Figura 1.23: Ejercicio 13

### 1.2.15. Ejercicio 14

Imprimir los números primos del 1 al 50

```
#!/bin/bash
#Program Ejercicio 14
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "N meros primos del 1 al 50:"
for (( num = 2; num <= 50; num++ )); do
    es_primo=1
    for (( i = 2; i <= num / 2; i++ )); do
        if (( num % i == 0 )); then
            es_primo=0
            break
        fi
    done
    if (( es_primo == 1 )); then
        echo $num
    fi
done

echo "I love Linux"
```

A terminal window with a dark purple background. It displays the output of a script: 'Números primos del 1 al 50:' followed by a list of prime numbers from 2 to 47. Below the list, it says 'I love Linux'. The prompt 'igucci@igucci-VirtualBox:~\$' is visible at the bottom with a white cursor.

```
Números primos del 1 al 50:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.24: Ejercicio 14

### 1.2.16. Ejercicio 15

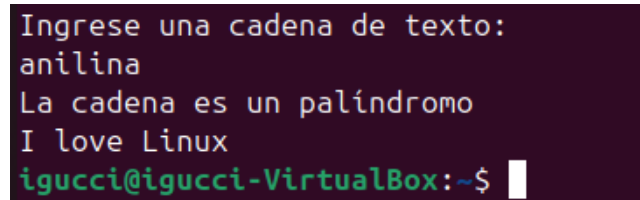
Verificar si una cadena es un palíndromo

```
#!/bin/bash
#Program Ejercicio 15
#Authors: Gustavo Aguas y Sebastian Paucar
#Date: 29-05-2023
echo "Ingrese una cadena de texto:"
read cadena

cadena_invertida=$(echo $cadena | rev)

if [ "$cadena" == "$cadena_invertida" ]; then
    echo "La cadena es un pal ndromo"
else
    echo "La cadena no es un pal ndromo"
fi

echo "I love Linux"
```

A terminal window with a dark purple background. It shows the execution of the script from Ejercicio 15. The user is prompted to enter a string, enters 'anilina', and the script outputs 'La cadena es un palíndromo'. Below that, it says 'I love Linux'. The prompt 'igucci@igucci-VirtualBox:~\$' is visible at the bottom with a white cursor.

```
Ingrese una cadena de texto:
anilina
La cadena es un palíndromo
I love Linux
igucci@igucci-VirtualBox:~$
```

Figura 1.25: Ejercicio 15

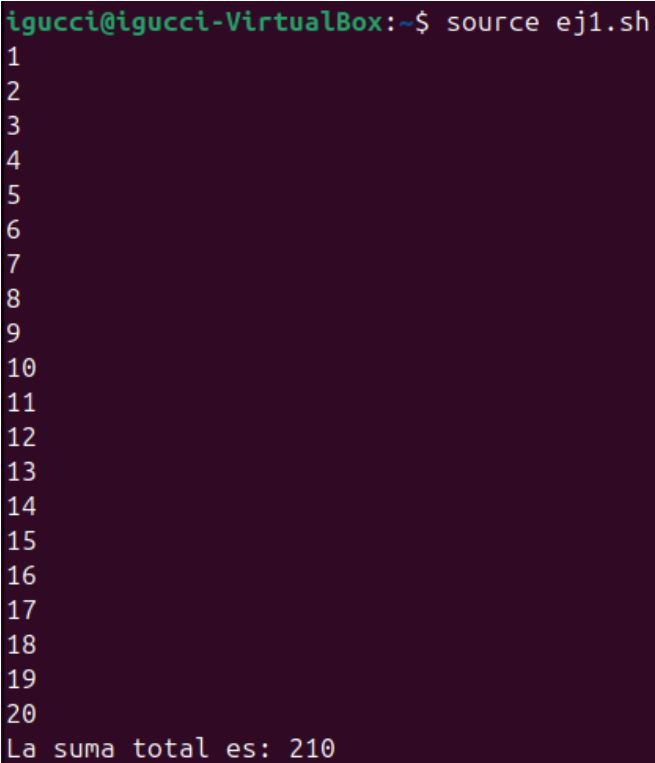
## 1.3. Tarea 3

### 1.3.1. Ejercicios de series

Realizar 10 ejercicios

### 1.3.2. Ejercicio 1

```
#!/bin/bash
# Program Ejercicio 1
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de n meros del 1 al 20 y su suma
  total
sum=0
for i in {1..20}; do
    echo $i
    sum=$((sum + i))
done
echo "La suma total es: $sum"
# I love Linux
```

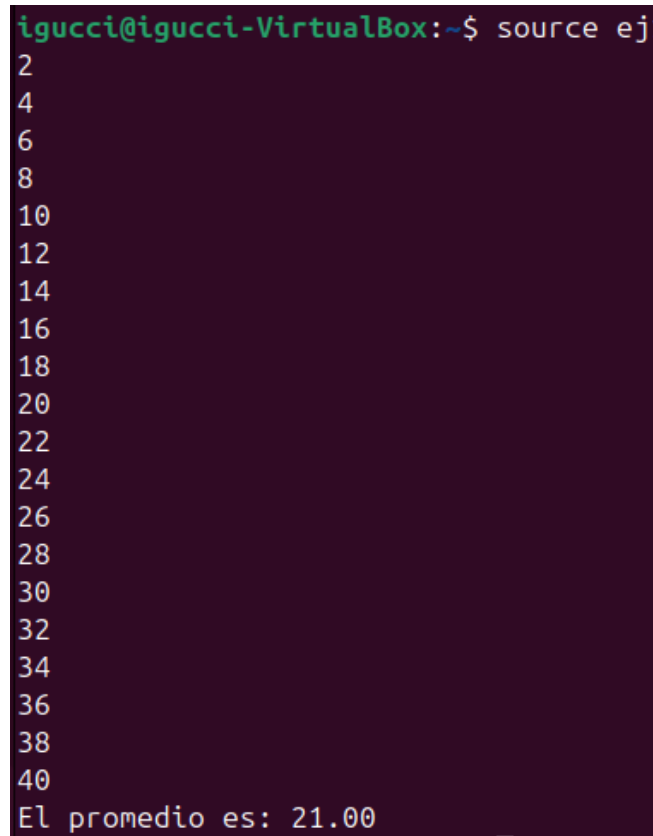


```
igucci@igucci-VirtualBox:~$ source ej1.sh
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
La suma total es: 210
```

Figura 1.26: Ejercicio 1

### 1.3.3. Ejercicio 2

```
#!/bin/bash
# Program Ejercicio 2
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de n meros pares del 2 al 40 y su
  promedio
sum=0
count=0
for i in {2..40..2}; do
    echo $i
    sum=$((sum + i))
    count=$((count + 1))
done
average=$(echo "scale=2; $sum / $count" | bc)
echo "El promedio es: $average"
# I love Linux
```



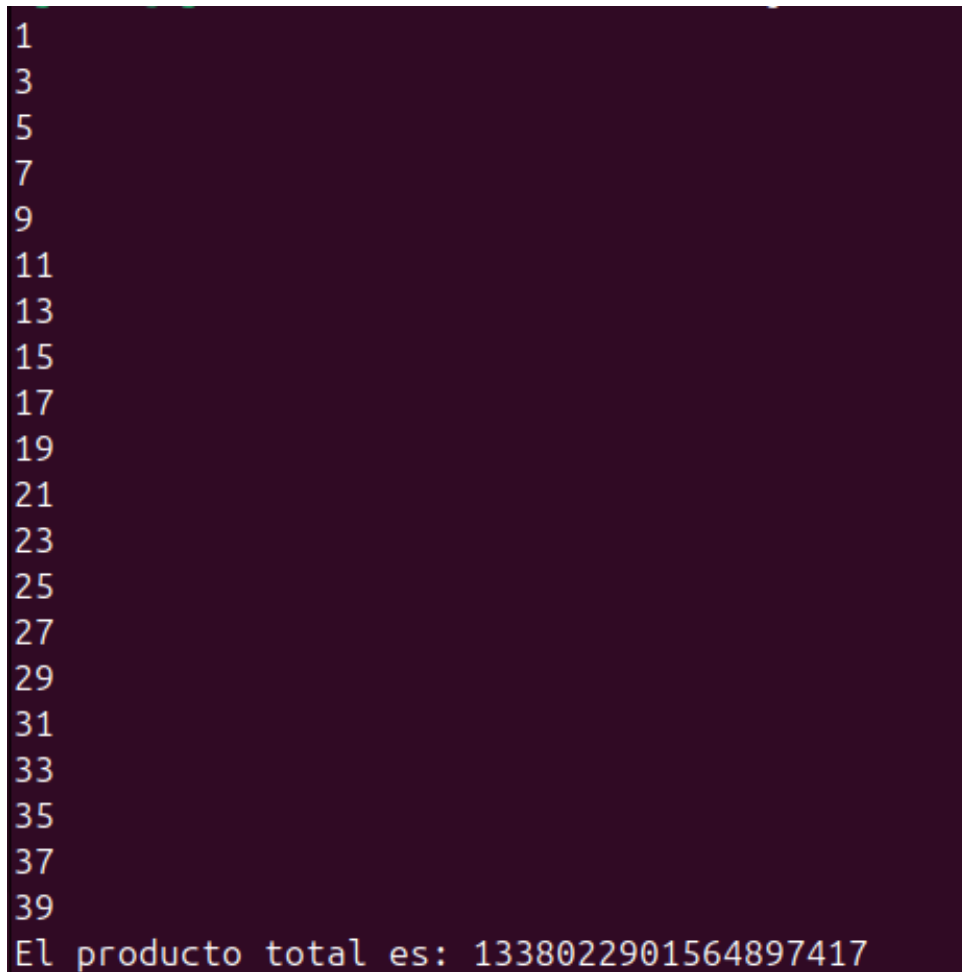
```
igucci@igucci-VirtualBox:~$ source ej
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
El promedio es: 21.00
```

Figura 1.27: Ejercicio 2

### 1.3.4. Ejercicio 3

```
#!/bin/bash
# Program Ejercicio 3
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de n meros impares del 1 al 39 y su
  producto total

product=1
for i in {1..39..2}; do
    echo $i
    product=$((product * i))
done
echo "El producto total es: $product"
#I love linux
```



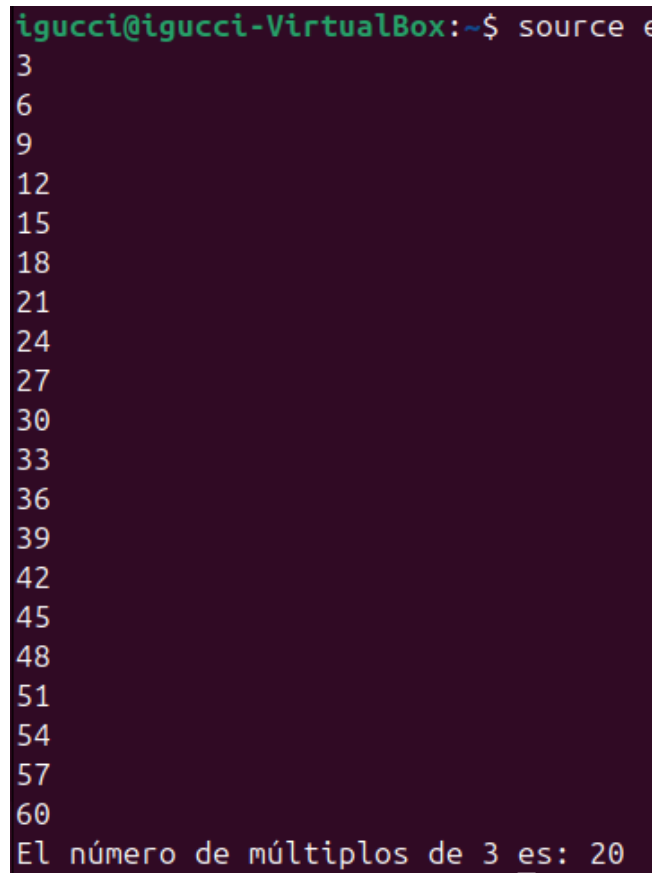
```
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
El producto total es: 1338022901564897417
```

Figura 1.28: Ejercicio 3

### 1.3.5. Ejercicio 4

```
#!/bin/bash
# Program Ejercicio 4
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de m ltiplos de 3 hasta 60 y contar
  cu ntos hay

count=0
for ((i=3; i<=60; i+=3)); do
    echo $i
    count=$((count + 1))
done
echo "El n mero de m ltiplos de 3 es: $count"
#I love linux
```



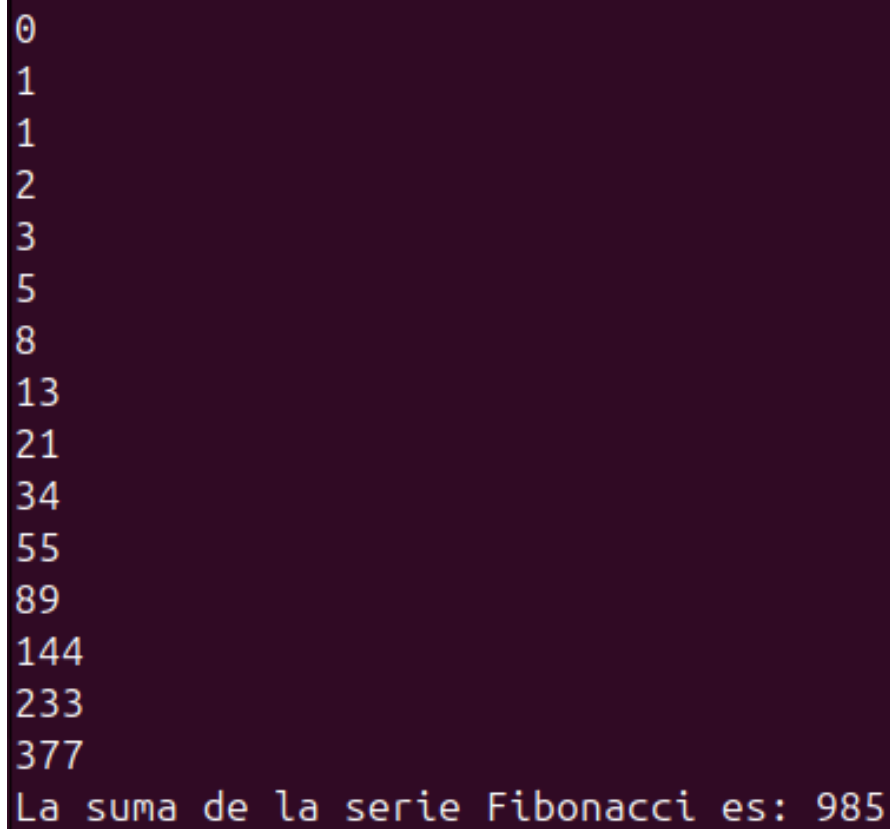
```
igucci@igucci-VirtualBox:~$ source e
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
El número de múltiplos de 3 es: 20
```

Figura 1.29: Ejercicio 4

### 1.3.6. Ejercicio 5

```
#!/bin/bash
# Program Ejercicio 5
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de Fibonacci (los primeros 15
  n meros) y su suma

a=0
b=1
sum=$a
echo $a
echo $b
for ((i=2; i<15; i++)); do
  c=$((a + b))
  echo $c
  sum=$((sum + c))
  a=$b
  b=$c
done
echo "La suma de la serie Fibonacci es: $sum"
#I love linux
```



```
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
La suma de la serie Fibonacci es: 985
```

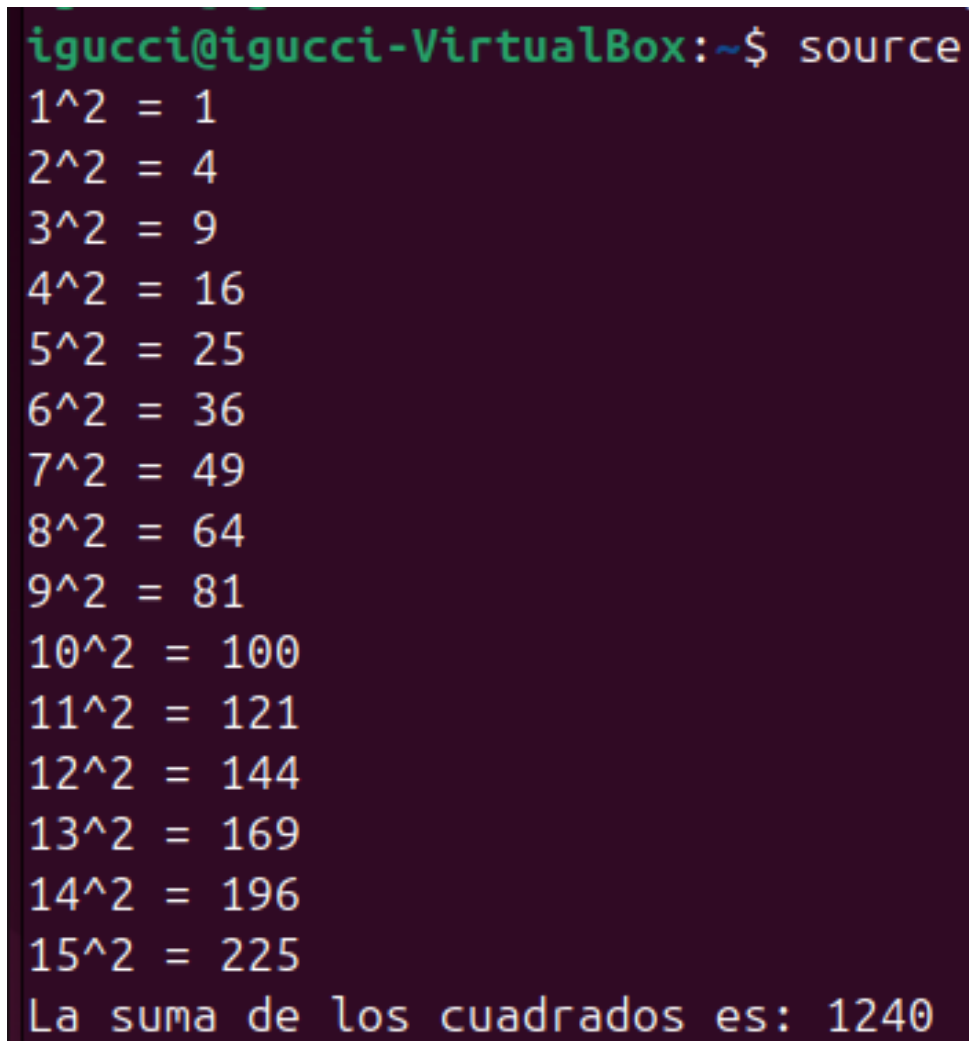
Figura 1.30: Ejercicio 5



### 1.3.7. Ejercicio 6

```
#!/bin/bash
# Program Ejercicio 6
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de cuadrados de los n meros del 1
    al 15 y su suma

sum=0
for i in {1..15}; do
    square=$(echo "$i^2" | bc)
    echo "$i^2 = $square"
    sum=$(echo "$sum + $square" | bc)
done
echo "La suma de los cuadrados es: $sum"
#I love linux
```



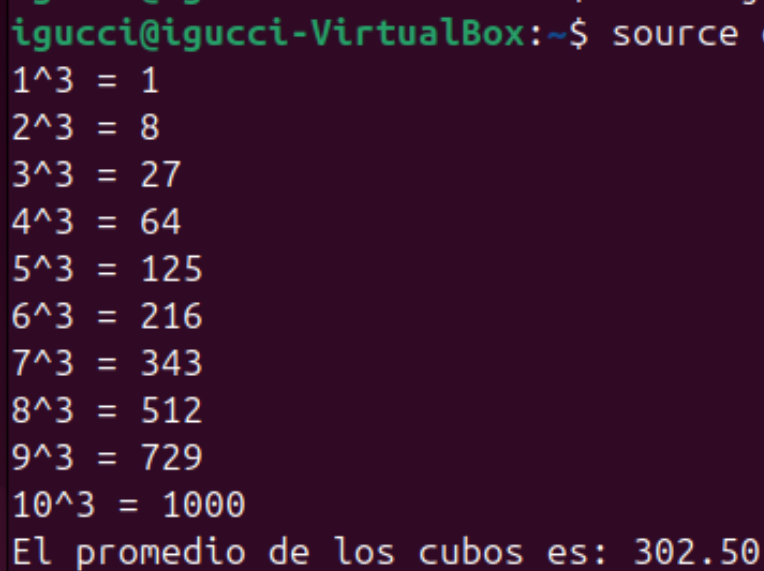
```
igucci@igucci-VirtualBox:~$ source
1^2 = 1
2^2 = 4
3^2 = 9
4^2 = 16
5^2 = 25
6^2 = 36
7^2 = 49
8^2 = 64
9^2 = 81
10^2 = 100
11^2 = 121
12^2 = 144
13^2 = 169
14^2 = 196
15^2 = 225
La suma de los cuadrados es: 1240
```

Figura 1.31: Ejercicio 6

### 1.3.8. Ejercicio 7

```
#!/bin/bash
# Program Ejercicio 7
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de cubos de los n meros del 1 al 10
  y su promedio

sum=0
count=0
for i in {1..10}; do
    cube=$(echo "$i^3" | bc)
    echo "$i^3 = $cube"
    sum=$(echo "$sum + $cube" | bc)
    count=$((count + 1))
done
average=$(echo "scale=2; $sum / $count" | bc)
echo "El promedio de los cubos es: $average"
#I love linux
```



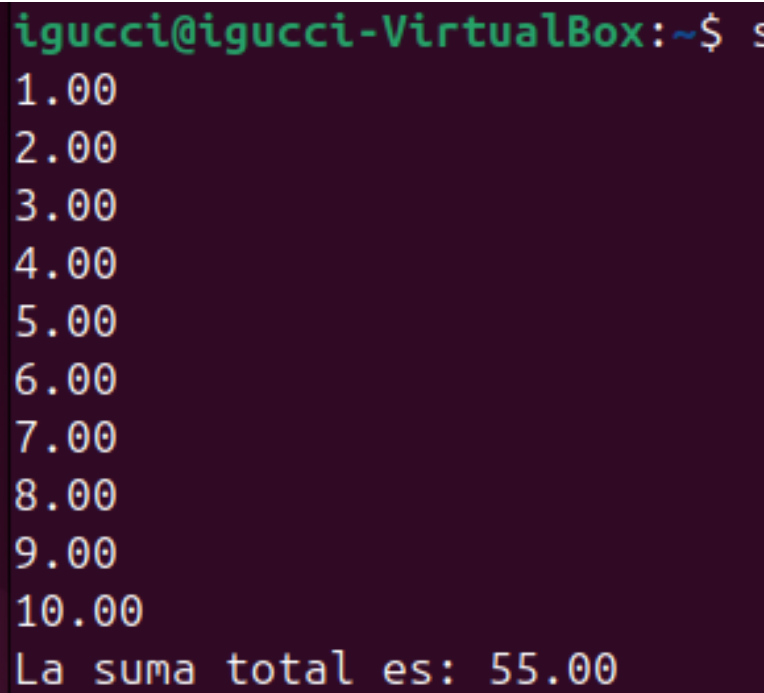
```
igucci@igucci-VirtualBox:~$ source e
1^3 = 1
2^3 = 8
3^3 = 27
4^3 = 64
5^3 = 125
6^3 = 216
7^3 = 343
8^3 = 512
9^3 = 729
10^3 = 1000
El promedio de los cubos es: 302.50
```

Figura 1.32: Ejercicio 7

### 1.3.9. Ejercicio 8

```
#!/bin/bash
# Program Ejercicio 8
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de los primeros 10 n meros con dos
    decimales y su suma

sum=0
for i in {1..10}; do
    value=$(echo "scale=2; $i/1" | bc)
    echo $value
    sum=$(echo "$sum + $value" | bc)
done
echo "La suma total es: $sum"
#I love linux
```



```
igucci@igucci-VirtualBox:~$ s
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
La suma total es: 55.00
```

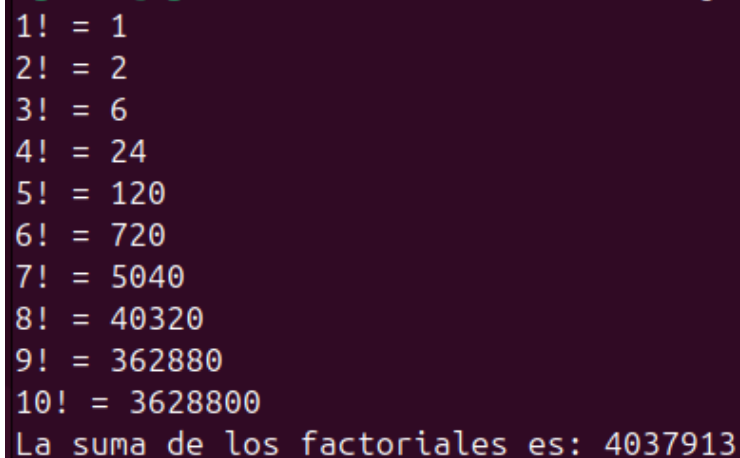
Figura 1.33: Ejercicio 8

**1.3.10. Ejercicio 9**

```
#!/bin/bash
# Program Ejercicio 9
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de los primeros 10 n meros
  factoriales y su suma

factorial() {
    if [ $1 -le 1 ]; then
        echo 1
    else
        echo "$1 * $(factorial $(( $1 - 1 )))" | bc
    fi
}

sum=0
for i in {1..10}; do
    fact=$(factorial $i)
    echo "$i! = $fact"
    sum=$(echo "$sum + $fact" | bc)
done
echo "La suma de los factoriales es: $sum"
#I love linux
```



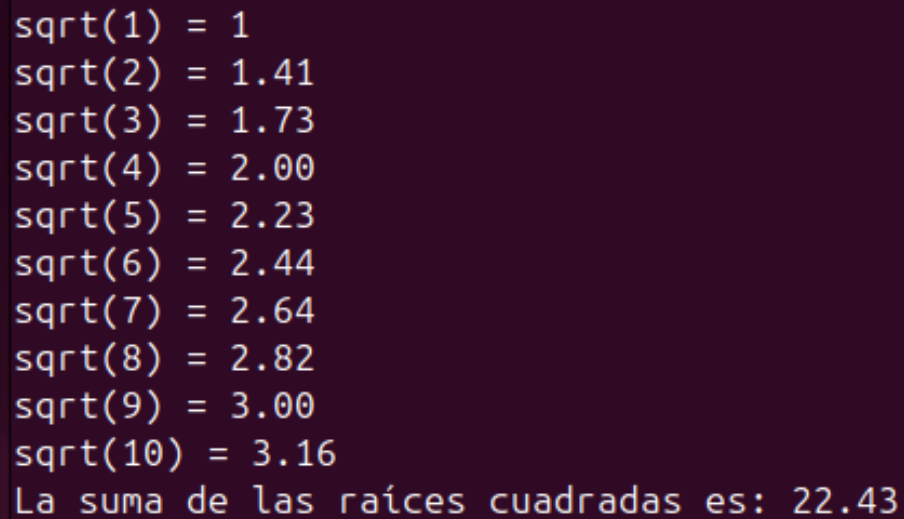
```
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
La suma de los factoriales es: 4037913
```

Figura 1.34: Ejercicio 9

**1.3.11. Ejercicio 10**

```
#!/bin/bash
# Program Ejercicio 10
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Serie de los primeros 10 n meros con
  ra z cuadrada y su suma

sum=0
for i in {1..10}; do
    sqrt=$(echo "scale=2; sqrt($i)" | bc)
    echo "sqrt($i) = $sqrt"
    sum=$(echo "$sum + $sqrt" | bc)
done
echo "La suma de las ra ces cuadradas es: $sum"
#I love linux
```



```
sqrt(1) = 1
sqrt(2) = 1.41
sqrt(3) = 1.73
sqrt(4) = 2.00
sqrt(5) = 2.23
sqrt(6) = 2.44
sqrt(7) = 2.64
sqrt(8) = 2.82
sqrt(9) = 3.00
sqrt(10) = 3.16
La suma de las raíces cuadradas es: 22.43
```

Figura 1.35: Ejercicio 10

## 1.4. Tarea 4

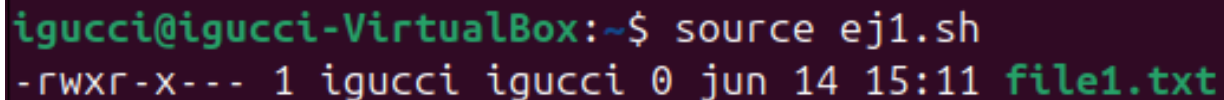
### 1.4.1. Permisos numericos

Realizar 10 ejercicios

### 1.4.2. Ejercicio 1

```
#!/bin/bash
# Program Ejercicio 1
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file1.txt y otorgarle
# permisos de lectura, escritura y ejecuci n al usuario; solo
# lectura y ejecuci n al grupo y sin permisos a otros:

touch file1.txt
chmod 750 file1.txt
ls -l file1.txt
# I love Linux
```



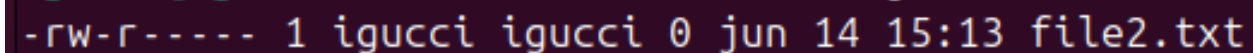
```
igucci@igucci-VirtualBox:~$ source ej1.sh
-rwxr-x--- 1 igucci igucci 0 jun 14 15:11 file1.txt
```

Figura 1.36: Ejercicio 1

### 1.4.3. Ejercicio 2

```
#!/bin/bash
# Program Ejercicio 2
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file2.txt y otorgarle
# permisos de lectura y escritura al usuario; solo lectura al grupo y
# sin permisos a otros:

touch file2.txt
chmod 640 file2.txt
ls -l file2.txt
# I love Linux
```



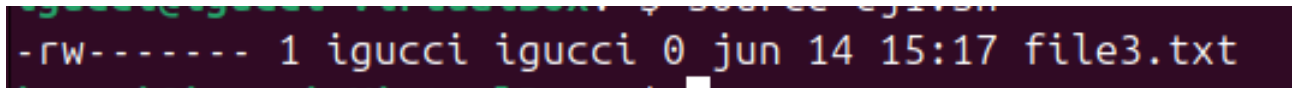
```
igucci@igucci-VirtualBox:~$ source ej2.sh
-rw-r----- 1 igucci igucci 0 jun 14 15:13 file2.txt
```

Figura 1.37: Ejercicio 2

### 1.4.4. Ejercicio 3

```
#!/bin/bash
# Program Ejercicio 3
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file3.txt y otorgarle
    permisos de lectura y escritura al usuario; sin permisos al grupo y
    otros:

touch file3.txt
chmod 600 file3.txt
ls -l file3.txt
#I love linux
```



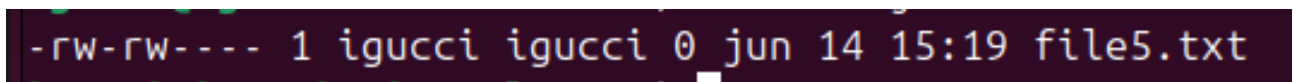
```
-rw----- 1 igucci igucci 0 jun 14 15:17 file3.txt
```

Figura 1.38: Ejercicio 3

#### 1.4.5. Ejercicio 4

```
#!/bin/bash
# Program Ejercicio 4
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file4.txt y otorgarle
    permisos de lectura, escritura y ejecuci n al usuario y grupo;
    solo lectura y ejecuci n a otros:

touch file4.txt
chmod 775 file4.txt
ls -l file4.txt
#I love linux
```



```
-rw-rw---- 1 igucci igucci 0 jun 14 15:19 file5.txt
```

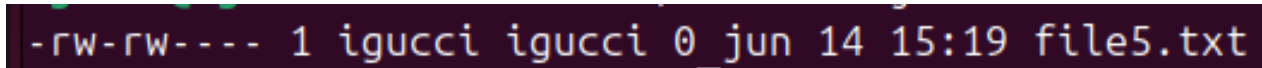
Figura 1.39: Ejercicio 4

#### 1.4.6. Ejercicio 5

```
#!/bin/bash
# Program Ejercicio 5
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file5.txt y otorgarle
    permisos de lectura y escritura al usuario y grupo; sin permisos a
    otros:

touch file5.txt
```

```
chmod 660 file5.txt
ls -l file5.txt
#I love linux
```



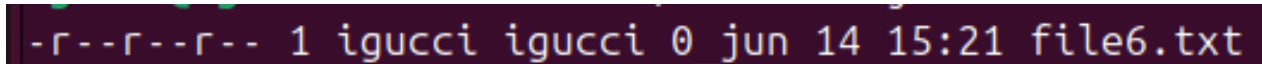
```
-rw-rw---- 1 igucci igucci 0 jun 14 15:19 file5.txt
```

Figura 1.40: Ejercicio 5

#### 1.4.7. Ejercicio 6

```
#!/bin/bash
# Program Ejercicio 6
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file6.txt y otorgarle
# permisos de solo lectura a todos:
bash

touch file6.txt
chmod 444 file6.txt
ls -l file6.txt
#I love linux
```



```
-r--r--r-- 1 igucci igucci 0 jun 14 15:21 file6.txt
```

Figura 1.41: Ejercicio 6

#### 1.4.8. Ejercicio 7

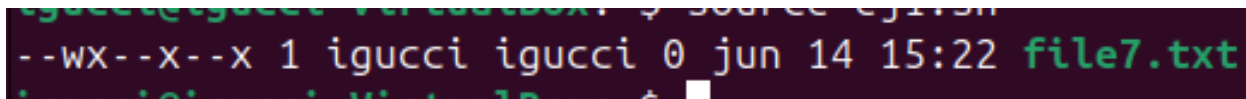
```
#!/bin/bash
# Program Ejercicio 7
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file7.txt y otorgarle
# permisos de escritura y ejecuci n al usuario; solo ejecuci n al
# grupo y otros:

touch file7.txt
chmod 311 file7.txt
ls -l file7.txt
#I love linux
```

#### 1.4.9. Ejercicio 8

```
#!/bin/bash
# Program Ejercicio 8
```



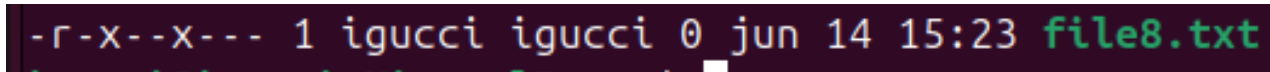


```
igucci@igucci-VirtualBox: ~$ source ej1.sh
--wx--x--x 1 igucci igucci 0 jun 14 15:22 file7.txt
```

Figura 1.42: Ejercicio 7

```
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file8.txt y otorgarle
  permisos de lectura y ejecuci n al usuario; solo ejecuci n al
  grupo y sin permisos a otros:
```

```
touch file8.txt
chmod 510 file8.txt
ls -l file8.txt
#I love linux
```



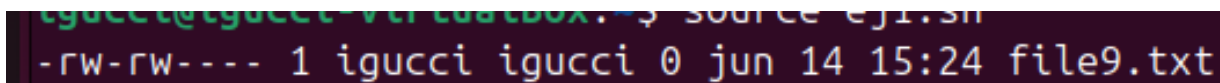
```
igucci@igucci-VirtualBox: ~$ source ej1.sh
-r-x--x--- 1 igucci igucci 0 jun 14 15:23 file8.txt
```

Figura 1.43: Ejercicio 8

#### 1.4.10. Ejercicio 9

```
#!/bin/bash
# Program Ejercicio 9
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file9.txt y otorgarle
  permisos de lectura y escritura al usuario; solo lectura y
  escritura al grupo y sin permisos a otros:
```

```
touch file9.txt
chmod 660 file9.txt
ls -l file9.txt
#I love linux
```



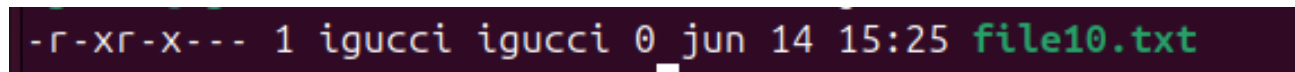
```
igucci@igucci-VirtualBox: ~$ source ej1.sh
-rw-rw---- 1 igucci igucci 0 jun 14 15:24 file9.txt
```

Figura 1.44: Ejercicio 9

**1.4.11. Ejercicio 10**

```
#!/bin/bash
# Program Ejercicio 10
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file10.txt y otorgarle
#   permisos de lectura y ejecuci n al usuario y grupo; sin permisos a
#   otros:

touch file10.txt
chmod 550 file10.txt
ls -l file10.txt
#I love linux
```



```
-r-xr-x--- 1 igucci igucci 0 jun 14 15:25 file10.txt
```

Figura 1.45: Ejercicio 10

## 1.5. Tarea 5

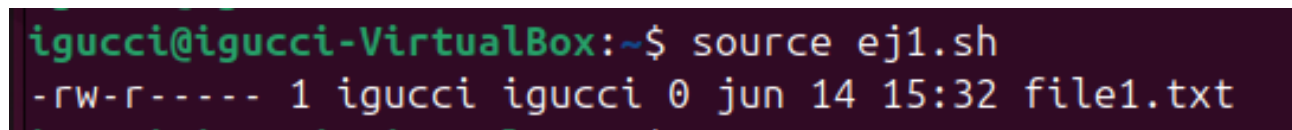
### 1.5.1. Permisos Simolicos

Realizar 10 ejercicios

### 1.5.2. Ejercicio 1

```
#!/bin/bash
# Program Ejercicio 1
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file1.txt y otorgarle
    permisos de lectura y escritura al usuario; solo lectura al grupo y
    sin permisos a otros:

touch file1.txt
chmod u=rw,g=r,o= file1.txt
ls -l file1.txt
# I love Linux
```



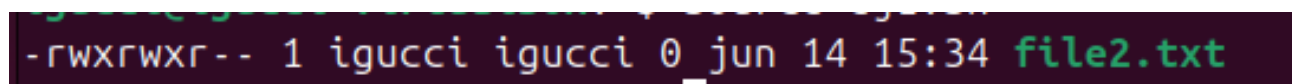
```
igucci@igucci-VirtualBox:~$ source ej1.sh
-rw-r----- 1 igucci igucci 0 jun 14 15:32 file1.txt
```

Figura 1.46: Ejercicio 1

### 1.5.3. Ejercicio 2

```
#!/bin/bash
# Program Ejercicio 2
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file2.txt y otorgarle
    permisos de lectura, escritura y ejecuci n al usuario y grupo;
    solo lectura a otros:

touch file2.txt
chmod u=rwx,g=rwx,o=r file2.txt
ls -l file2.txt
# I love Linux
```



```
igucci@igucci-VirtualBox:~$ source ej2.sh
-rwxrwxr-- 1 igucci igucci 0 jun 14 15:34 file2.txt
```

Figura 1.47: Ejercicio 2

**1.5.4. Ejercicio 3**

```
#!/bin/bash
# Program Ejercicio 3
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file3.txt y otorgarle
    permisos de solo lectura al usuario; sin permisos al grupo y otros:

touch file3.txt
chmod u=r,g=,o= file3.txt
ls -l file3.txt
#I love linux
```

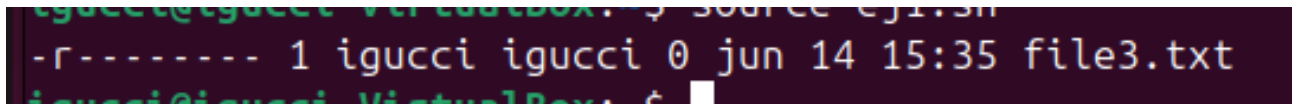


Figura 1.48: Ejercicio 3

**1.5.5. Ejercicio 4**

```
#!/bin/bash
# Program Ejercicio 4
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file4.txt y otorgarle
    permisos de lectura y ejecuci n al usuario y grupo; sin permisos a
    otros:

touch file4.txt
chmod u=rx,g=rx,o= file4.txt
ls -l file4.txt
#I love linux
```

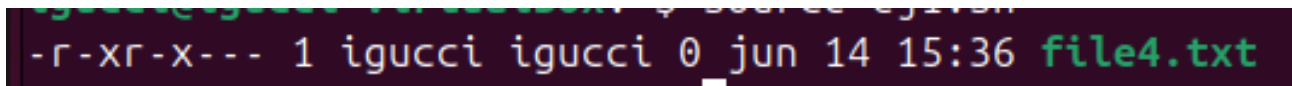


Figura 1.49: Ejercicio 4

**1.5.6. Ejercicio 5**

```
#!/bin/bash
# Program Ejercicio 5
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file5.txt y otorgarle
    permisos de escritura y ejecuci n al usuario; solo ejecuci n al
    grupo y otros:
```

```
touch file5.txt
chmod u=wx,g=x,o=x file5.txt
ls -l file5.txt
#I love linux
```

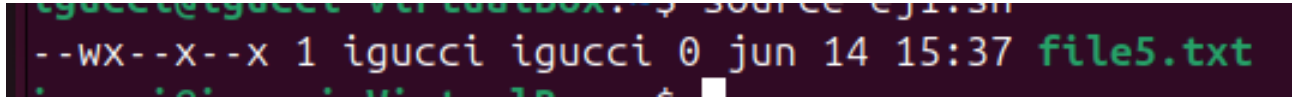


Figura 1.50: Ejercicio 5

### 1.5.7. Ejercicio 6

```
#!/bin/bash
# Program Ejercicio 6
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file6.txt y otorgarle
# permisos de lectura y escritura al usuario y grupo; sin permisos a
# otros:
```

```
touch file6.txt
chmod u=rw,g=rw,o= file6.txt
ls -l file6.txt
#I love linux
```

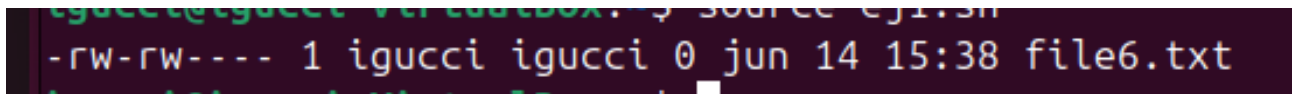
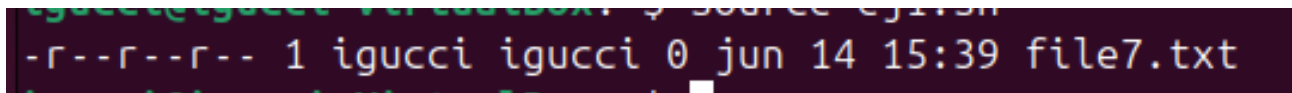


Figura 1.51: Ejercicio 6

### 1.5.8. Ejercicio 7

```
#!/bin/bash
# Program Ejercicio 7
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file7.txt y otorgarle
# permisos de solo lectura a todos:
```

```
touch file7.txt
chmod u=r,g=r,o=r file7.txt
ls -l file7.txt
#I love linux
```



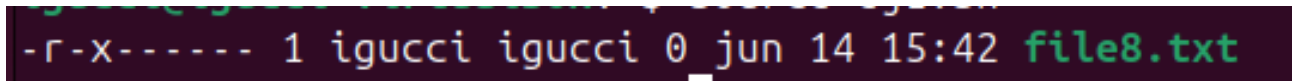
```
-r--r--r-- 1 igucci igucci 0 jun 14 15:39 file7.txt
```

Figura 1.52: Ejercicio 7

### 1.5.9. Ejercicio 8

```
#!/bin/bash
# Program Ejercicio 8
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file8.txt y otorgarle
#   permisos de lectura y ejecuci n al usuario; sin permisos al grupo
#   y otros:
```

```
touch file8.txt
chmod u=rx,g=,o= file8.txt
ls -l file8.txt
#I love linux
```



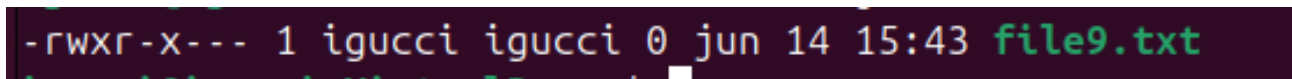
```
-r-x----- 1 igucci igucci 0 jun 14 15:42 file8.txt
```

Figura 1.53: Ejercicio 8

### 1.5.10. Ejercicio 9

```
#!/bin/bash
# Program Ejercicio 9
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file9.txt y otorgarle
#   permisos de lectura, escritura y ejecuci n al usuario; solo
#   lectura y ejecuci n al grupo; sin permisos a otros:
```

```
touch file9.txt
chmod u=rwx,g=rx,o= file9.txt
ls -l file9.txt
#I love linux
```



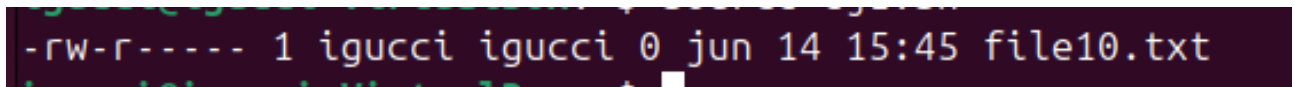
```
-rwxr-x--- 1 igucci igucci 0 jun 14 15:43 file9.txt
```

Figura 1.54: Ejercicio 9

**1.5.11. Ejercicio 10**

```
#!/bin/bash
# Program Ejercicio 10
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Crear un archivo file10.txt y otorgarle
#   permisos de lectura y escritura al usuario; solo lectura al grupo;
#   sin permisos a otros:

touch file10.txt
chmod u=rw,g=r,o= file10.txt
ls -l file10.txt
#I love linux
```



```
-rw-r----- 1 igucci igucci 0 jun 14 15:45 file10.txt
```

Figura 1.55: Ejercicio 10

## 1.6. Tarea 6

### 1.6.1. Ejercicios con Funciones

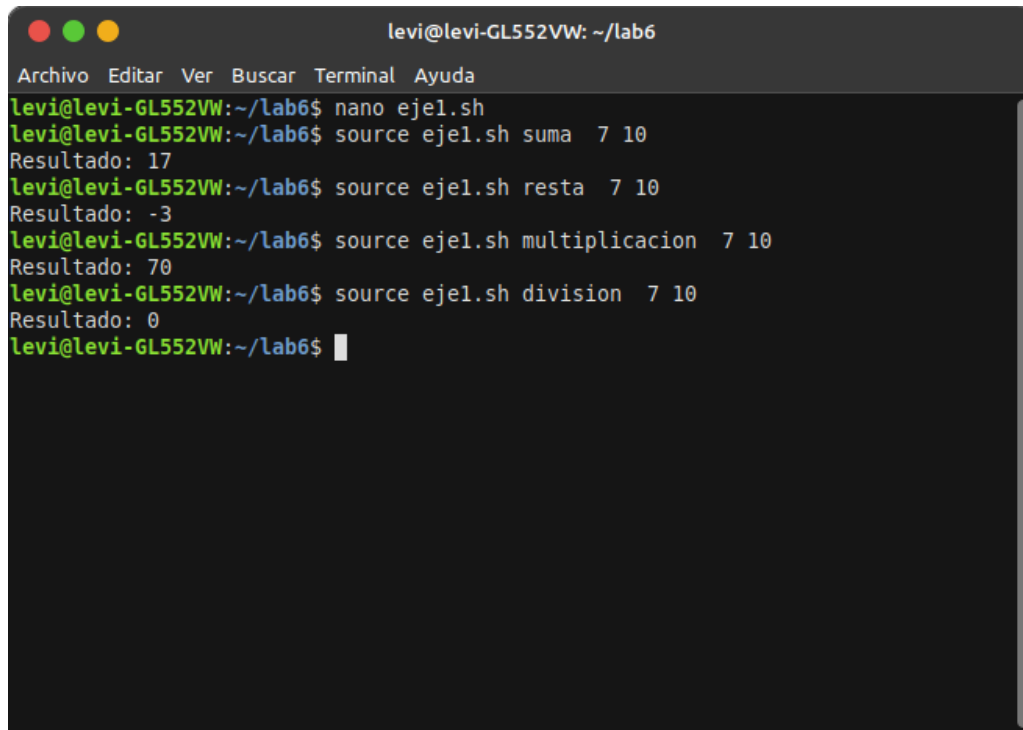
Realizar 10 ejercicios

### 1.6.2. Ejercicio 1

```
#!/bin/bash
# Program Ejercicio 1
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script realiza operaciones
# matematicas basicas (suma, resta, multiplicacion, division)
# segun los parametros proporcionados.

function calculadora() {
    local operacion=$1
    local num1=$2
    local num2=$3
    case $operacion in
        "suma")
            echo "Resultado: $((num1 + num2))"
            ;;
        "resta")
            echo "Resultado: $((num1 - num2))"
            ;;
        "multiplicacion")
            echo "Resultado: $((num1 * num2))"
            ;;
        "division")
            if [ $num2 -ne 0 ]; then
                echo "Resultado: $((num1 / num2))"
            else
                echo "Error: Divisi n por cero"
            fi
            ;;
        *)
            echo "Operaci n no v lida"
            ;;
    esac
}
calculadora $1 $2 $3
#I love linux
```





```
levi@levi-GL552VW: ~/lab6
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~/lab6$ nano ejel.sh
levi@levi-GL552VW:~/lab6$ source ejel.sh suma 7 10
Resultado: 17
levi@levi-GL552VW:~/lab6$ source ejel.sh resta 7 10
Resultado: -3
levi@levi-GL552VW:~/lab6$ source ejel.sh multiplicacion 7 10
Resultado: 70
levi@levi-GL552VW:~/lab6$ source ejel.sh division 7 10
Resultado: 0
levi@levi-GL552VW:~/lab6$
```

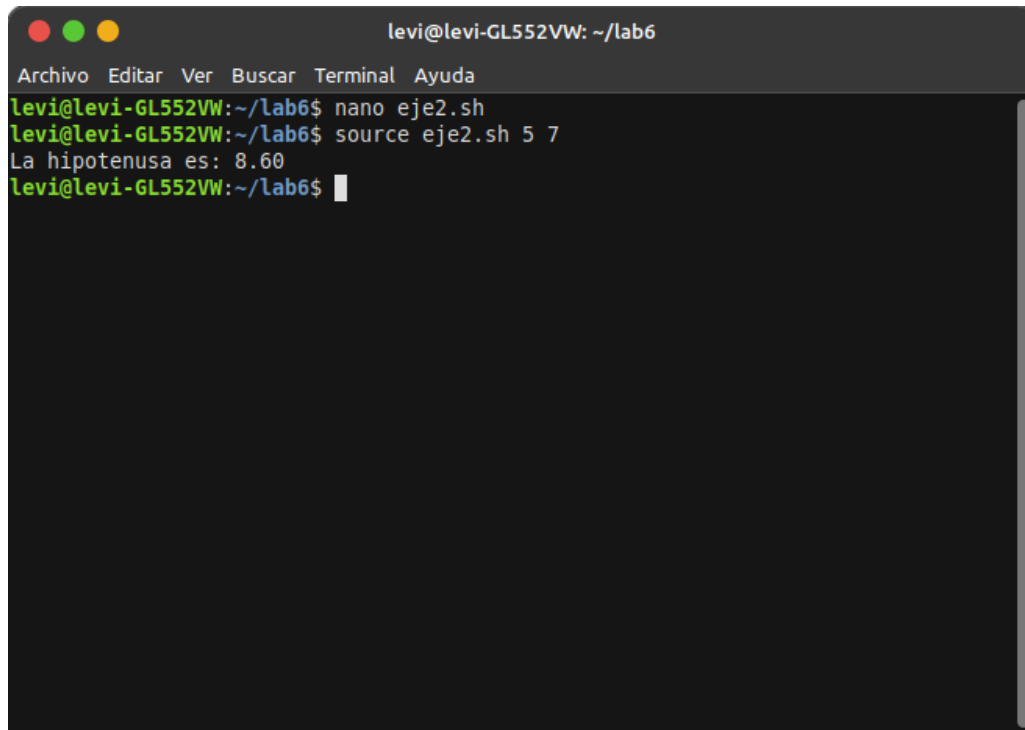
Figura 1.56: compilación del ejercicio 1

### 1.6.3. Ejercicio 2

```
#!/bin/bash
# Program Ejercicio 2
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script calcula la hipotenusa de un
# tri ngulo rect ngulo dados los catetos.

function calcular_hipotenusa() {
    local cateto1=$1
    local cateto2=$2
    local hipotenusa=$(echo "scale=2; sqrt($cateto1^2 + $cateto2^2)" |
        bc)
    echo "La hipotenusa es: $hipotenusa"
}

calcular_hipotenusa $1 $2
#I love linux
```

A terminal window titled 'levi@levi-GL552VW: ~/lab6' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
levi@levi-GL552VW:~/lab6$ nano eje2.sh
levi@levi-GL552VW:~/lab6$ source eje2.sh 5 7
La hipotenusa es: 8.60
levi@levi-GL552VW:~/lab6$
```

Figura 1.57: compilación del ejercicio 2

#### 1.6.4. Ejercicio 3

```
#!/bin/bash
# Program Ejercicio 3
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script convierte una cadena de texto
# pasada como par metro a may sculas.

function convertir_mayusculas() {
    local texto=$1
    echo "${texto^^}"
}

convertir_mayusculas "$1"

#I love linux
```

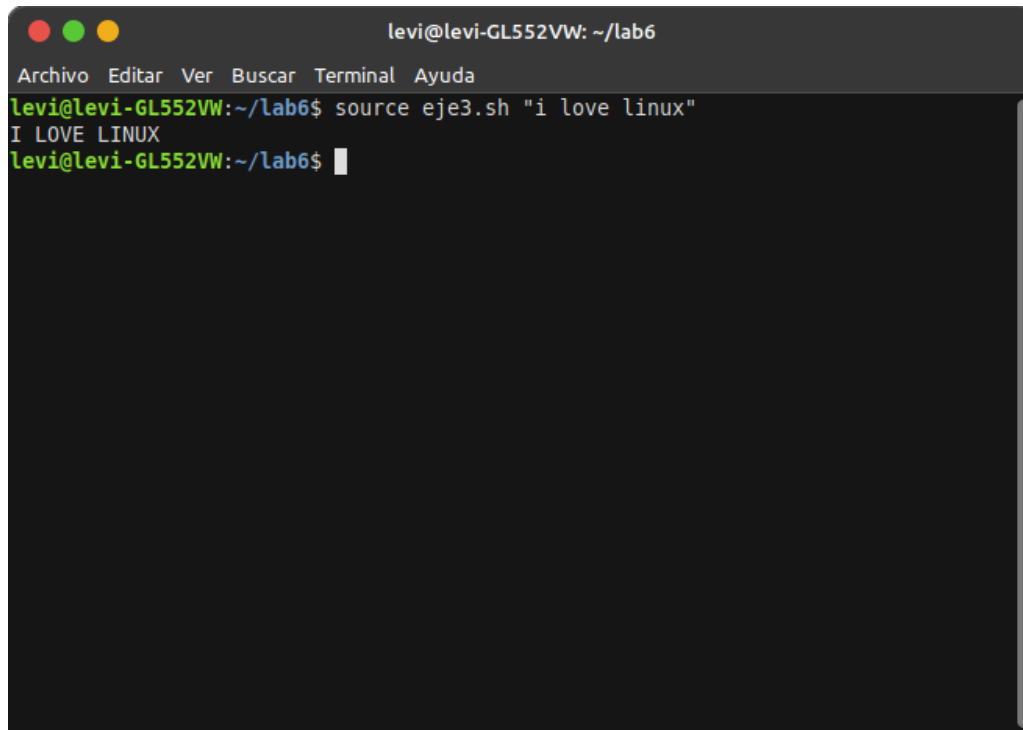


Figura 1.58: compilación del ejercicio 3

### 1.6.5. Ejercicio 4

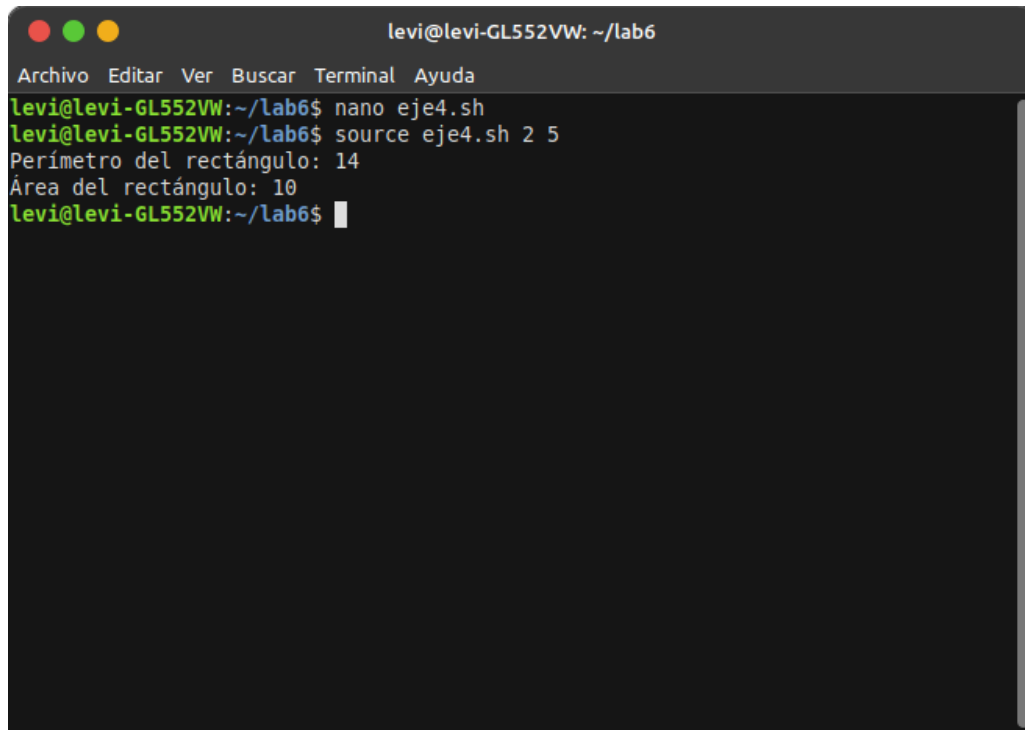
```
#!/bin/bash
# Program Ejercicio 4
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script calcula el per metro y el
#   rea de un rect ngulo dados su longitud y anchura.

function calcular_rectangulo() {
    local longitud=$1
    local anchura=$2

    local perimetro=$((2 * (longitud + anchura)))
    local area=$((longitud * anchura))

    echo "Per metro del rect ngulo: $perimetro"
    echo " rea del rect ngulo: $area"
}

calcular_rectangulo $1 $2
```

A terminal window titled 'levi@levi-GL552VW: ~/lab6' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
levi@levi-GL552VW:~/lab6$ nano eje4.sh
levi@levi-GL552VW:~/lab6$ source eje4.sh 2 5
Perímetro del rectángulo: 14
Área del rectángulo: 10
levi@levi-GL552VW:~/lab6$
```

Figura 1.59: compilación del ejercicio 4

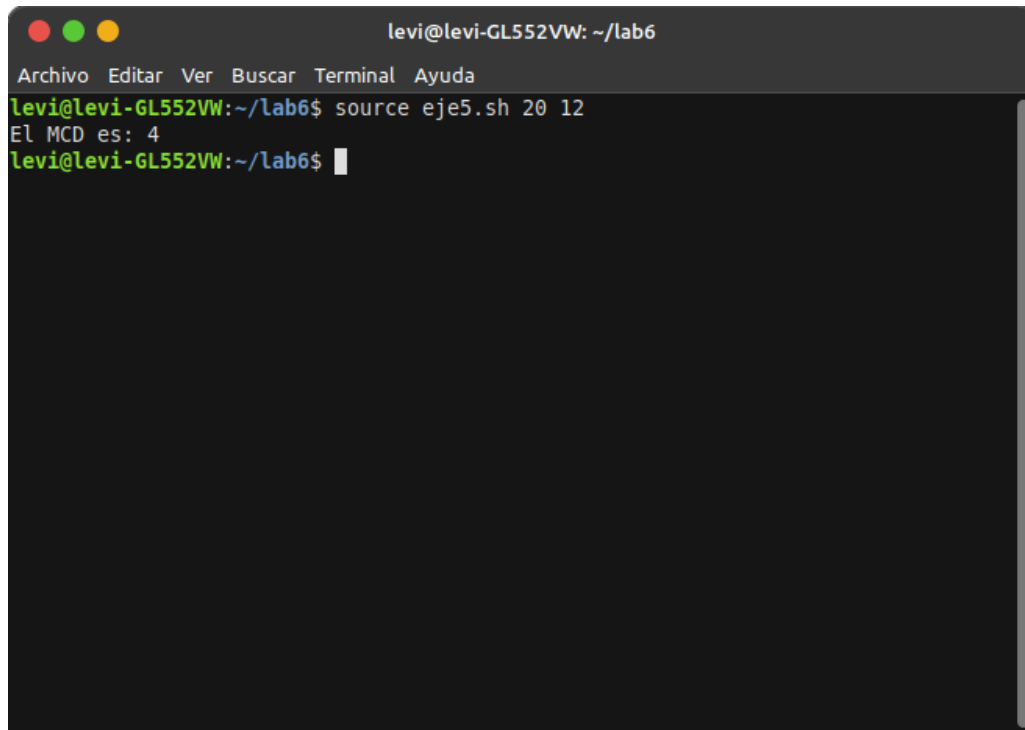
### 1.6.6. Ejercicio 5

```
#!/bin/bash
# Program Ejercicio 5
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script encuentra el máximo común
# divisor (MCD) de dos números dados.

function mcd() {
    local a=$1
    local b=$2
    while [ $b -ne 0 ]; do
        local temp=$b
        b=$((a % b))
        a=$temp
    done
    echo "El MCD es: $a"
}

mcd $1 $2

#I love linux
```

A terminal window titled 'levi@levi-GL552VW: ~/lab6'. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the command 'source eje5.sh 20 12' being executed, followed by the output 'El MCD es: 4'. The prompt 'levi@levi-GL552VW: ~/lab6\$' is visible at the bottom.

```
levi@levi-GL552VW: ~/lab6$ source eje5.sh 20 12
El MCD es: 4
levi@levi-GL552VW: ~/lab6$
```

Figura 1.60: compilación del ejercicio 5

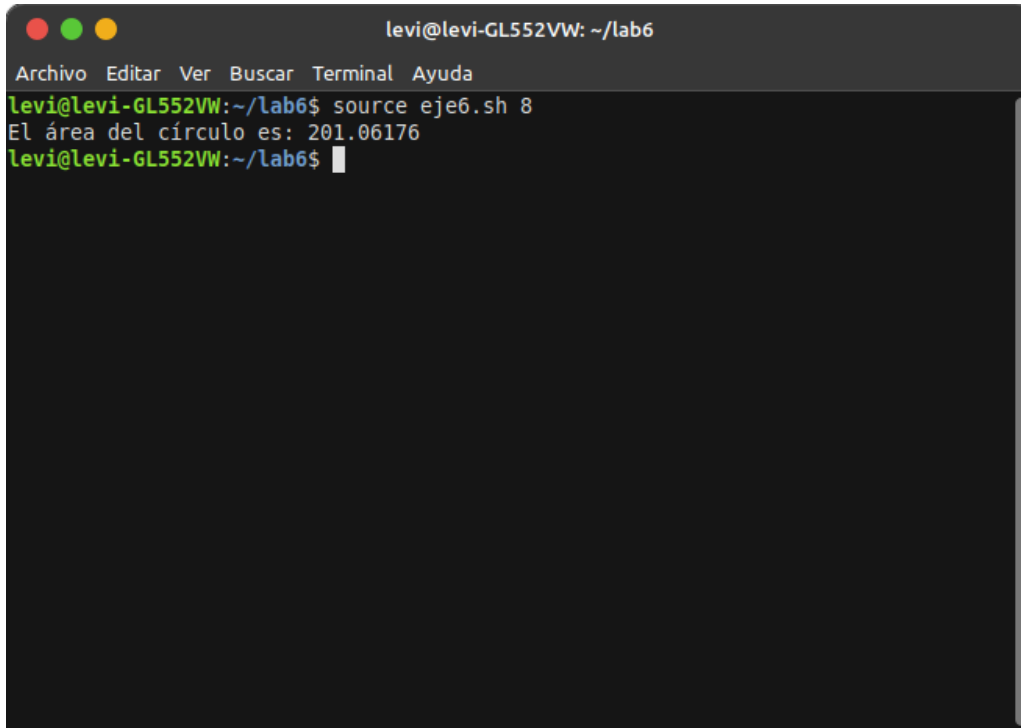
### 1.6.7. Ejercicio 6

```
#!/bin/bash
# Program Ejercicio 6
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script calcula el area de un
# circulo dado su radio.

function area_circulo() {
    local radio=$1
    local area=$(echo "scale=2; 3.14159 * $radio^2" | bc)
    echo "El area del circulo es: $area"
}

calcular_area_circulo $1

#I love linux
```

A terminal window titled 'levi@levi-GL552VW: ~/lab6' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the command 'source eje6.sh 8' being executed, followed by the output 'El área del círculo es: 201.06176'. The prompt 'levi@levi-GL552VW:~/lab6\$' is visible at the bottom.

```
levi@levi-GL552VW:~/lab6$ source eje6.sh 8
El área del círculo es: 201.06176
levi@levi-GL552VW:~/lab6$
```

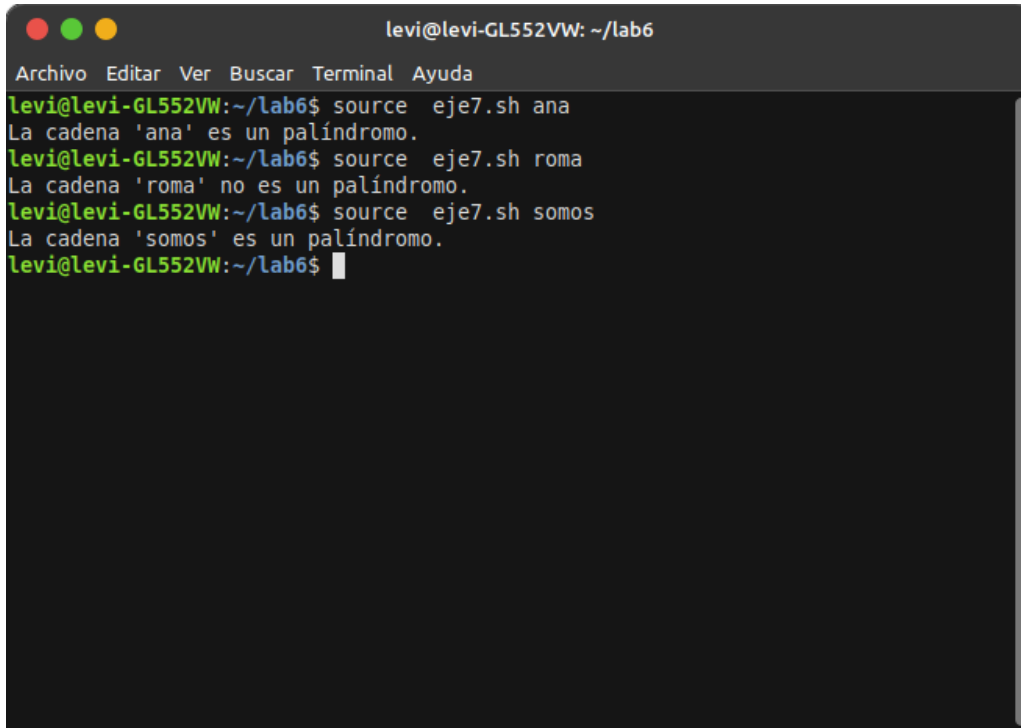
Figura 1.61: compilación del ejercicio 6

### 1.6.8. Ejercicio 7

```
#!/bin/bash
# Program Ejercicio 7
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script verifica si una cadena de
# texto es un pal ndromo.
function es_palindromo() {
    local cadena=$1
    local invertida=$(echo $cadena | rev)
    if [ "$cadena" == "$invertida" ]; then
        echo "La cadena '$cadena' es un pal ndromo."
    else
        echo "La cadena '$cadena' no es un pal ndromo."
    fi
}

es_palindromo $1

#I love linux
```



```
levi@levi-GL552VW: ~/lab6
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~/lab6$ source eje7.sh ana
La cadena 'ana' es un palíndromo.
levi@levi-GL552VW:~/lab6$ source eje7.sh roma
La cadena 'roma' no es un palíndromo.
levi@levi-GL552VW:~/lab6$ source eje7.sh somos
La cadena 'somos' es un palíndromo.
levi@levi-GL552VW:~/lab6$
```

Figura 1.62: compilación del ejercicio 7

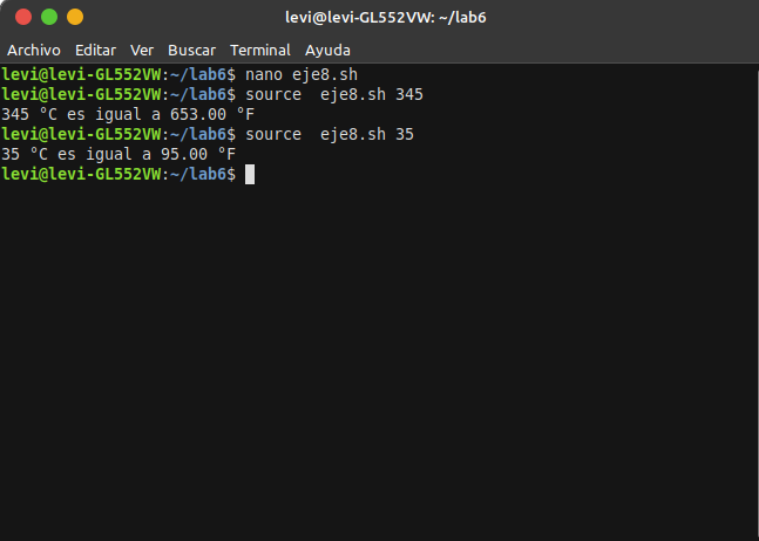
### 1.6.9. Ejercicio 8

```
#!/bin/bash
# Program Ejercicio 8
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script convierte una temperatura dada
# en grados Celsius a grados Fahrenheit.

function celsius_a_fahrenheit() {
    local celsius=$1
    local fahrenheit=$(echo "scale=2; ($celsius * 9/5) + 32" | bc)
    echo "$celsius C es igual a $fahrenheit F "
}

celsius_a_fahrenheit $1

#I love linux
```



```
levi@levi-GL552VW: ~/lab6
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~/lab6$ nano eje8.sh
levi@levi-GL552VW:~/lab6$ source eje8.sh 345
345 °C es igual a 653.00 °F
levi@levi-GL552VW:~/lab6$ source eje8.sh 35
35 °C es igual a 95.00 °F
levi@levi-GL552VW:~/lab6$
```

Figura 1.63: compilación del ejercicio 8

### 1.6.10. Ejercicio 9

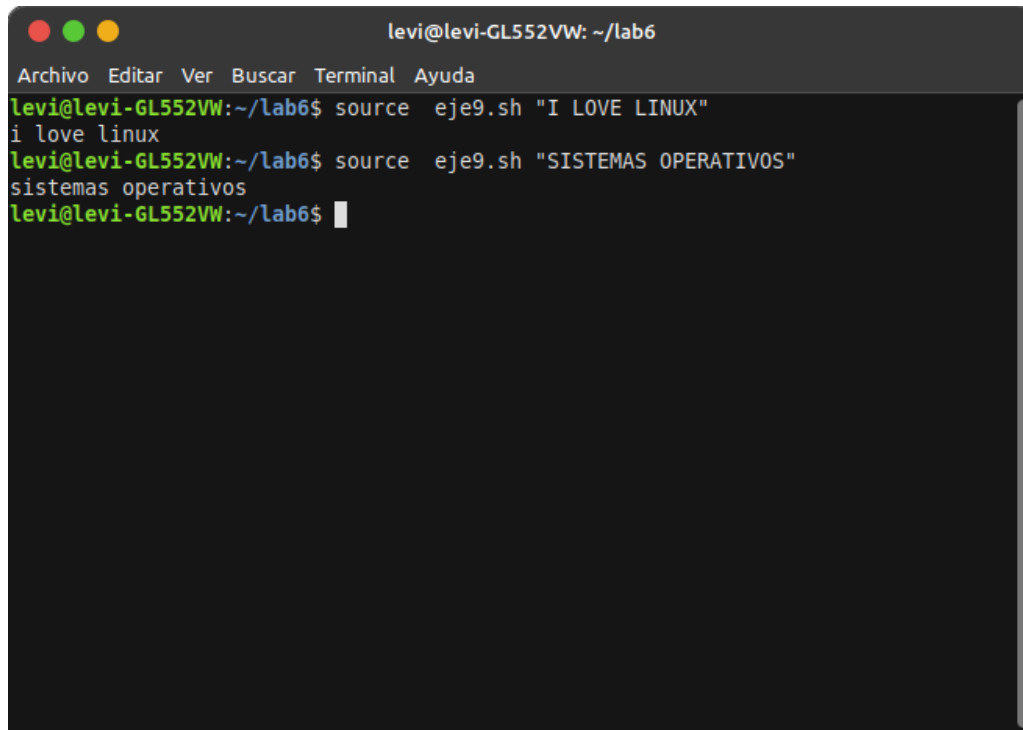
```
#!/bin/bash
# Program Ejercicio 9
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script convierte todas las letras
# may sculas en una cadena a min sculas.

function minusculas() {
    local texto=$1
    echo "${texto,,}"
}

Minusculas "$1"

#I love linux
```





```
levi@levi-GL552VW: ~/lab6
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
levi@levi-GL552VW:~/lab6$ source  eje9.sh "I LOVE LINUX"
i love linux
levi@levi-GL552VW:~/lab6$ source  eje9.sh "SISTEMAS OPERATIVOS"
sistemas operativos
levi@levi-GL552VW:~/lab6$
```

Figura 1.64: compilación del ejercicio 9

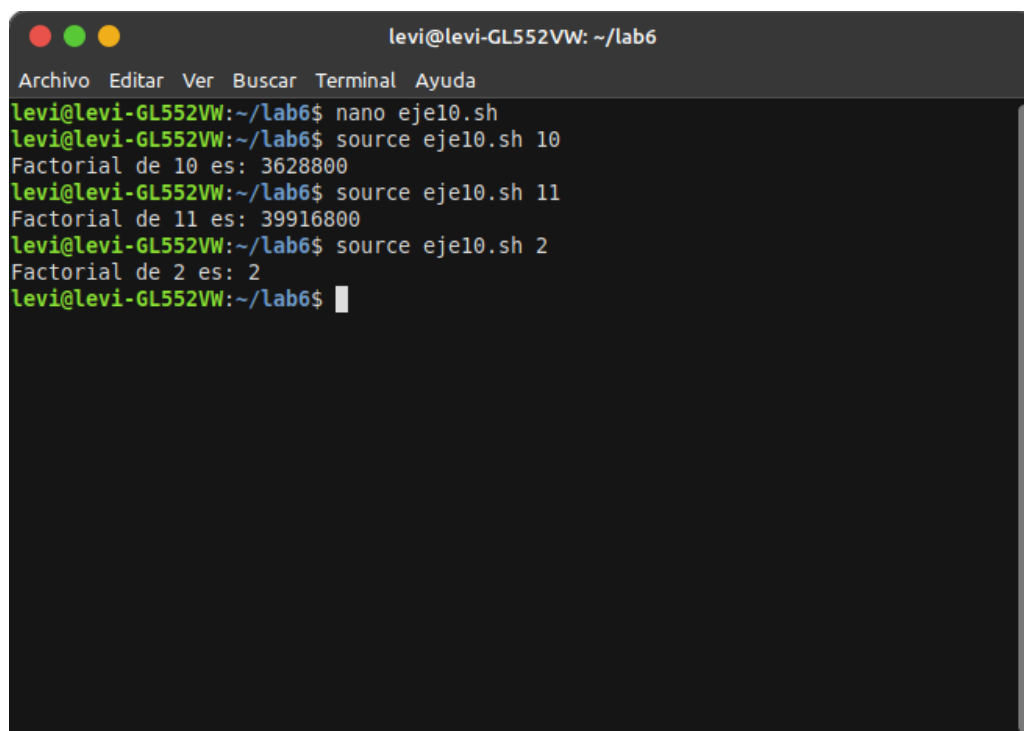
### 1.6.11. Ejercicio 10

```
#!/bin/bash
# Program Ejercicio 10
# Authors: Gustavo Aguas y Sebastian Paucar
# Explicacion del programa: Este script calcula el factorial de un
# n mero pasado como par metro.

function factorial() {
    local numero=$1
    local resultado=1
    for ((i=1; i<=numero; i++)); do
        resultado=$((resultado * i))
    done
    echo "Factorial de $numero es: $resultado"
}

factorial $1

#I love linux
```



```
levi@levi-GL552VW: ~/lab6
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
levi@levi-GL552VW:~/lab6$ nano eje10.sh
levi@levi-GL552VW:~/lab6$ source eje10.sh 10
Factorial de 10 es: 3628800
levi@levi-GL552VW:~/lab6$ source eje10.sh 11
Factorial de 11 es: 39916800
levi@levi-GL552VW:~/lab6$ source eje10.sh 2
Factorial de 2 es: 2
levi@levi-GL552VW:~/lab6$
```

Figura 1.65: compilación del ejercicio 10