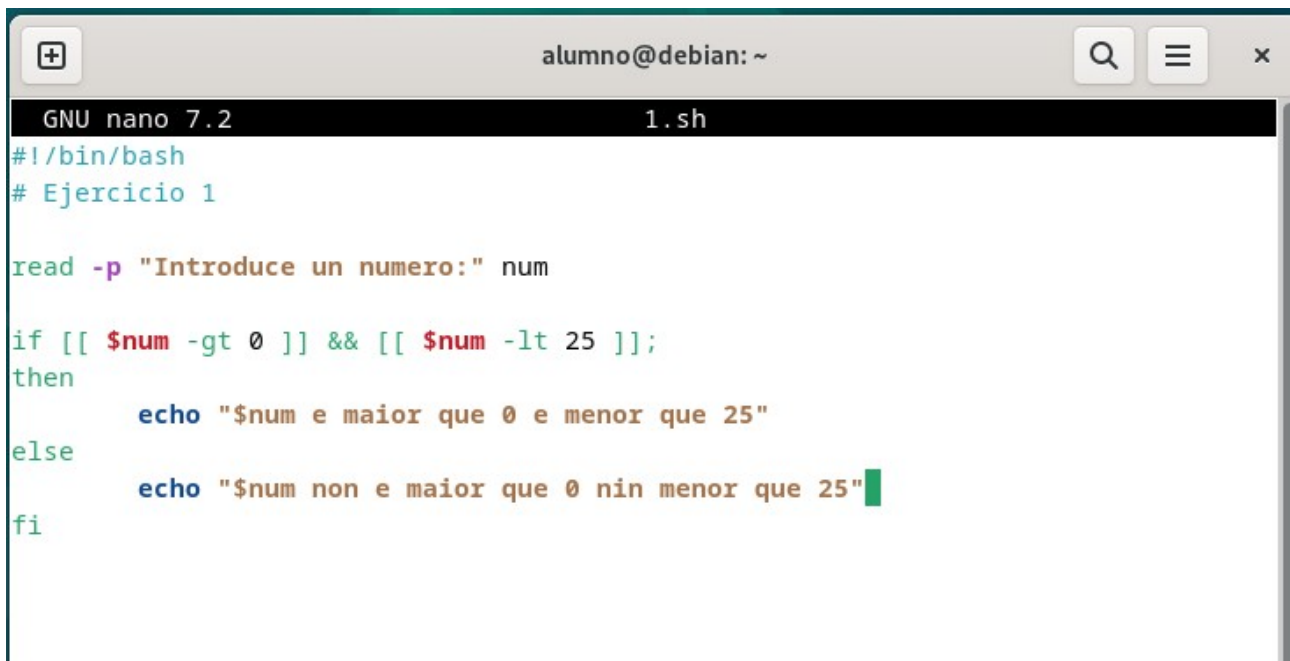


1. Realiza un script al que se le envíe un solo parámetro y comprobar que es mayor que 0 y menor que 25.



```
alumno@debian: ~  
GNU nano 7.2 1.sh  
#!/bin/bash  
# Ejercicio 1  
  
read -p "Introduce un numero:" num  
  
if [[ $num -gt 0 ]] && [[ $num -lt 25 ]];  
then  
    echo "$num e maior que 0 e menor que 25"  
else  
    echo "$num non e maior que 0 nin menor que 25"  
fi
```

2. Realiza un script que reciba dos números enteros como parámetros, comprueba que el primero es mayor que el segundo y posteriormente muestra los números del intervalo.



```
alumno@debian: ~  
GNU nano 7.2 2.sh  
#!/bin/bash  
# Ejercicio 2  
  
# Definir la función para mostrar el intervalo  
intervalo() {  
    local numero1=$1  
    local numero2=$2  
  
    if [ "$numero1" -gt "$numero2" ]; then  
        echo "El $numero1 es mayor que el $numero2"  
    fi  
  
    for (( i=numero2+1; i<numero1; i++ )); do  
        echo "$i"  
    done  
}  
  
read -p "Introduzca un numero:" numero1  
read -p "Introduzca otro numero:" numero2  
  
intervalo "$numero1" "$numero2"
```

3. Crea una función que indique si un número entero es primo o no. El número se pasa como parámetro a la función.



The screenshot shows a terminal window titled 'alumno@debian: ~'. Inside, the GNU nano 7.2 editor is open, editing a file named '3.sh'. The script is a Bash function named 'primo' that checks if a number is prime. It uses conditional statements to handle numbers 1 and 2, and a loop for numbers greater than 2. The script prompts the user to enter a number and then calls the 'primo' function with that number.

```
GNU nano 7.2 3.sh
#!/bin/bash
# Ejercicio 3

primo() {
    local numero=$1

    if [ "$numero" -le 1 ]; then
        echo "El número $numero no es primo."
        return
    fi

    if [ "$numero" -eq 2 ]; then
        echo "El número $numero es primo."
        return
    fi

    for (( i=2; i<numero; i++ )); do
        if [ $(numero % i) -eq 0 ]; then
            echo "El número $numero no es primo."
            return
        fi
    done

    echo "El número $numero es primo."
}

read -p "Introduzca un numero:" numero

primo "$numero"
```

4. Realiza un script con las siguientes opciones: (el menú se mostrará hasta que se pulse la opción 5)

1. Muestra el calendario
2. Muestra el número de palabras y de líneas que tiene un fichero (el nombre del fichero debe de ser introducido por teclado)
3. Copia un archivo a un directorio destino (tanto el archivo como el directorio destion se introducen por teclado)
4. Renombra un fichero existente. Tanto el nombre del fichero existente como el nombre del nuevo fichero deben de ser pasados como parámetros (el nombre del nuevo fichero no debe de existir en el sistema)
- 5.Salir

```
GNU nano 7.2          alumno@debian: ~
# ./bin/bash          4.sh
# Ejercicio 4

mostrar_calendario() {
    cal
}

contar_palabras_lineas() {
    echo "Introduce el nombre del fichero: "
    read archivo
    wc $archivo
}

copiar_archivo() {
    echo "Introduce el nombre del archivo a copiar: "
    read archivo
    echo "Introduce el directorio de destino: "
    read directorio
    cp $archivo $directorio
}

renombrar_archivo() {
    echo -n "Introduce el nombre del archivo existente: "
    read archivo
    echo -n "Introduce el nuevo nombre para el archivo: "
    read nuevo_nombre
    mv $archivo $nuevo_nombre
}

while true; do
    clear
    echo "1) Mostrar el calendario"
    echo "2) Mostrar el número de palabras y líneas de un fichero"
    echo "3) Copiar un archivo a un directorio destino"
    echo "4) Renombrar un archivo existente"
    echo "5) Salir"
    echo "Elige una opción: "
    read opcion

```

```
read opcion

if [[ $opcion -eq 1 ]]; then
    mostrar_calendario
elif [[ $opcion -eq 2 ]]; then
    contar_palabras_lineas
elif [[ $opcion -eq 3 ]]; then
    copiar_archivo
elif [[ $opcion -eq 4 ]]; then
    renombrar_archivo
elif [[ $opcion -eq 5 ]]; then
    exit 0
else
    echo "Opción no válida. Intenta de nuevo."
fi
done
```