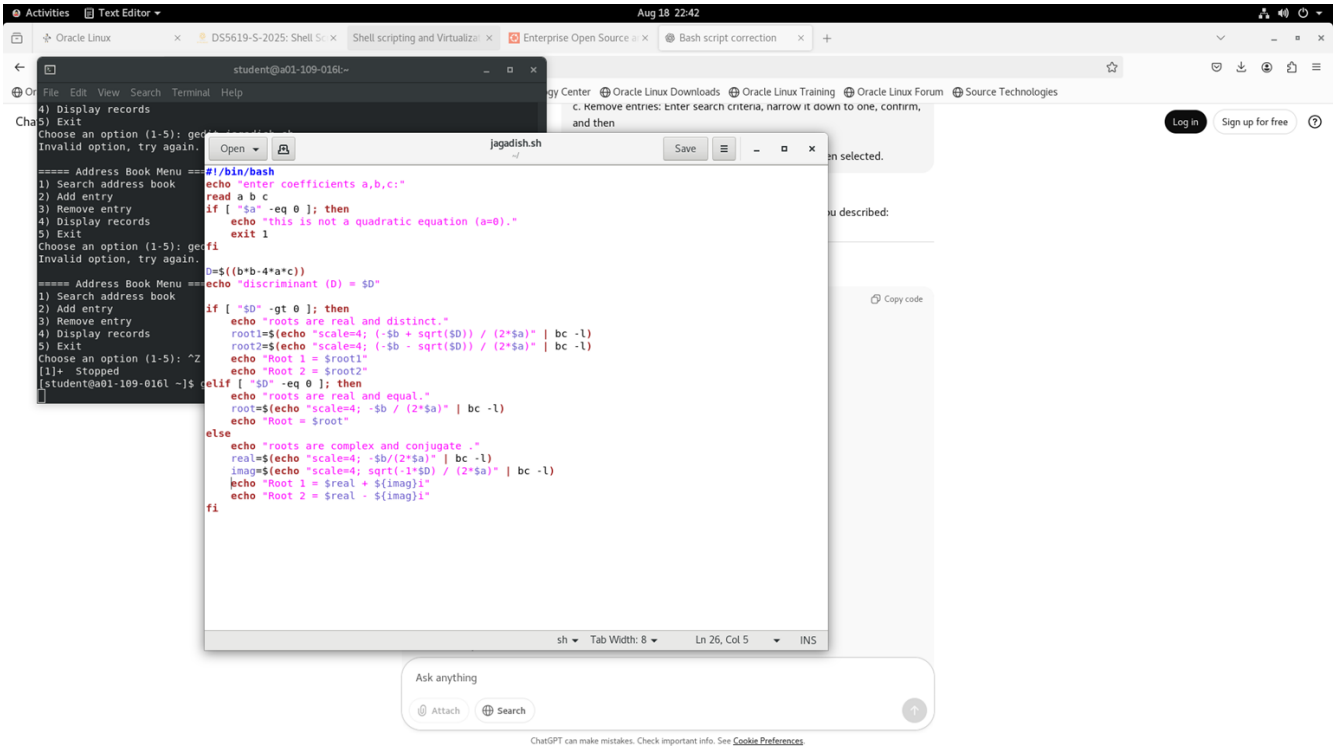
Shell scripting and Virtualization 142502007

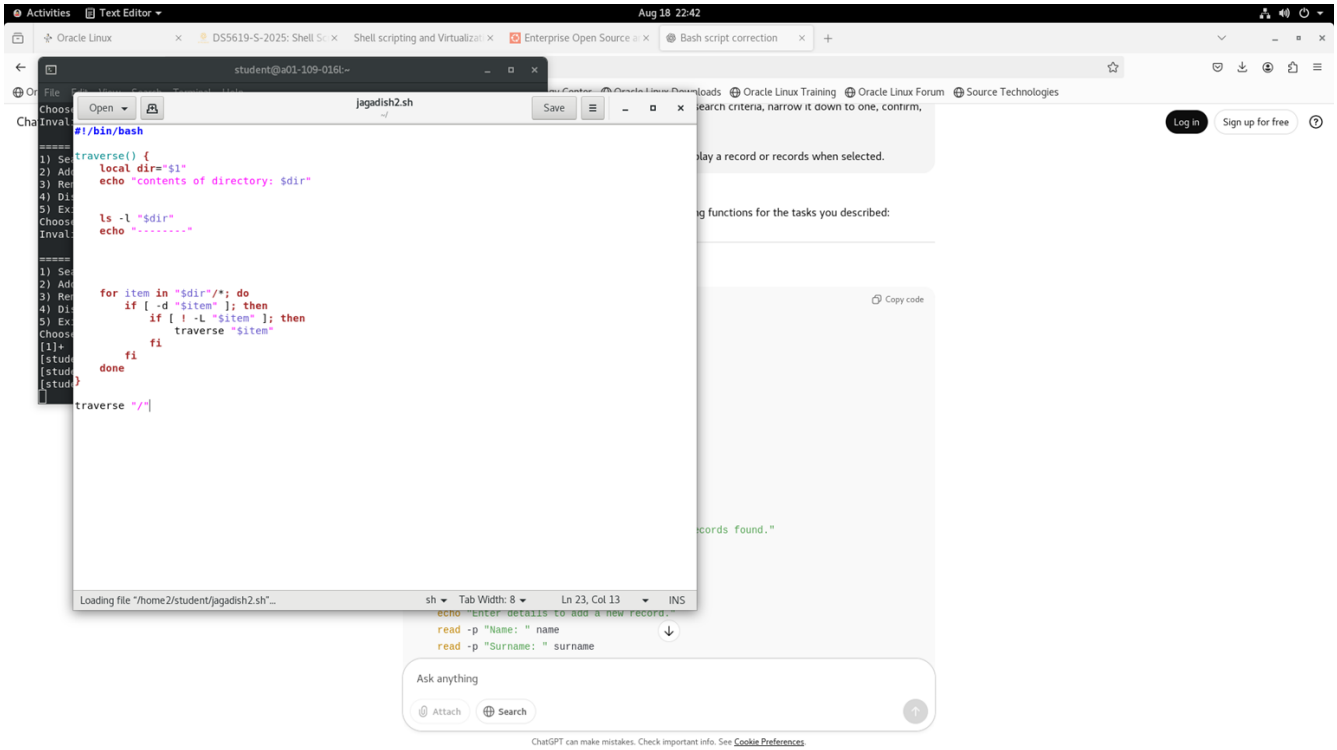
1. Write a shell script to find the roots of the quadratic equation - 2 Marks




```
#!/bin/bash
echo "enter coefficients a,b,c:"
read a b c
if [ "$a" -eq 0 ]; then
   echo "this is not a quadratic equation (a=0)."
   exit 1
fi

D=$((b*b-4*a*c))
echo "discriminant (D) = $D"

if [ "$D" -gt 0 ]; then
   echo "roots are real and distinct."
   root1=$(echo "scale=4; (-$b + sqrt($D)) / (2*$a)" | bc -l)
   root2=$(echo "scale=4; (-$b - sqrt($D)) / (2*$a)" | bc -l)
   echo "Root 1 = $root1"
   echo "Root 2 = $root2"
elif [ "$D" -eq 0 ]; then
   echo "roots are real and equal."
   root=$(echo "scale=4; -$b / (2*$a)" | bc -l)
   echo "Root = $root"
else
   echo "roots are complex and conjugate ."
   real=$(echo "scale=4; -$b/(2*$a)" | bc -l)
   imag=$(echo "scale=4; sqrt(-1*$D) / (2*$a)" | bc -l)
```

```
    echo "Root 1 = $real + ${imag}i"
    echo "Root 2 = $real - ${imag}i"
fi
```

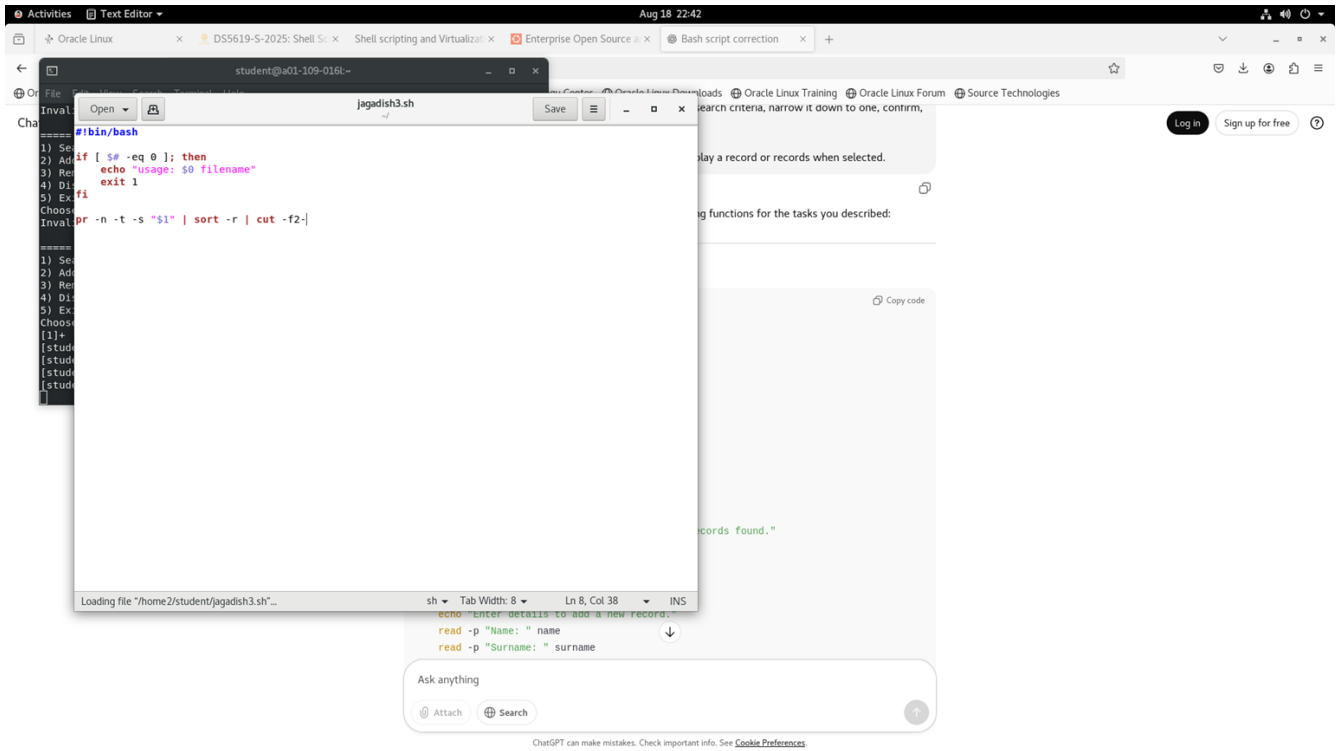
2.Write a shell script traverse through a filesystem tree. The script will start from the root file system “/” and traverse each subdirectory and list the contents of the directories.




```
#!/bin/bash

traverse() {
    local dir="$1"
    echo "contents of directory: $dir"


    ls -l "$dir"
    echo "--------"



    for item in "$dir"/*; do
        if [ -d "$item" ]; then
            if [ ! -L "$item" ]; then
                traverse "$item"
            fi
```

```
        fi
    done
}

traverse "/"
```

3.Use the commands pr, sort, and cut to read a text file in reverse order. For instance, let the content of the file (file.txt) is
One
Two
Three
Then, your script should display the contents of file.txt in reverse order:
Three
Two
One




```
#!bin/bash

if [ $# -eq 0 ]; then
    echo "usage: $0 filename"
    exit 1
fi

pr -n -t -s "$1" | sort -r | cut -f2-
```

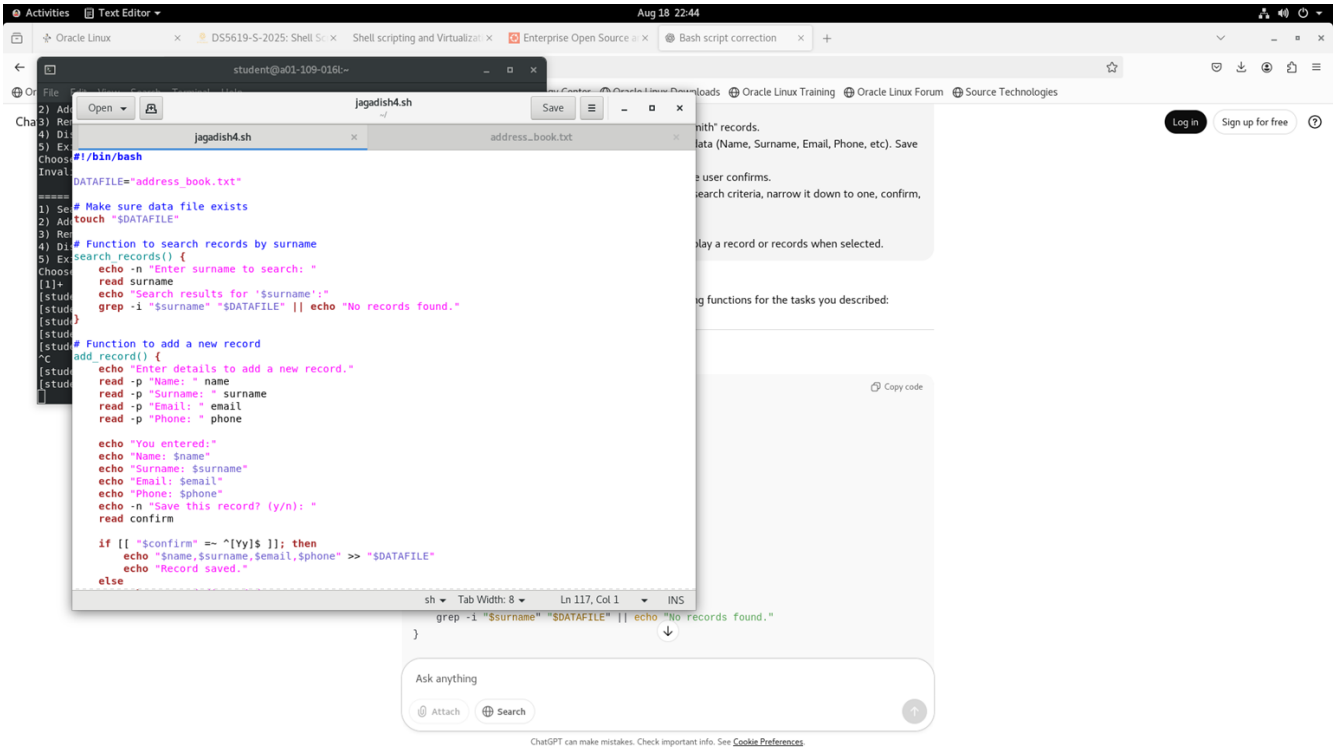
4.Create an address book program. It should use functions to perform the required tasks. It should be menu-based, allowing you the options of:

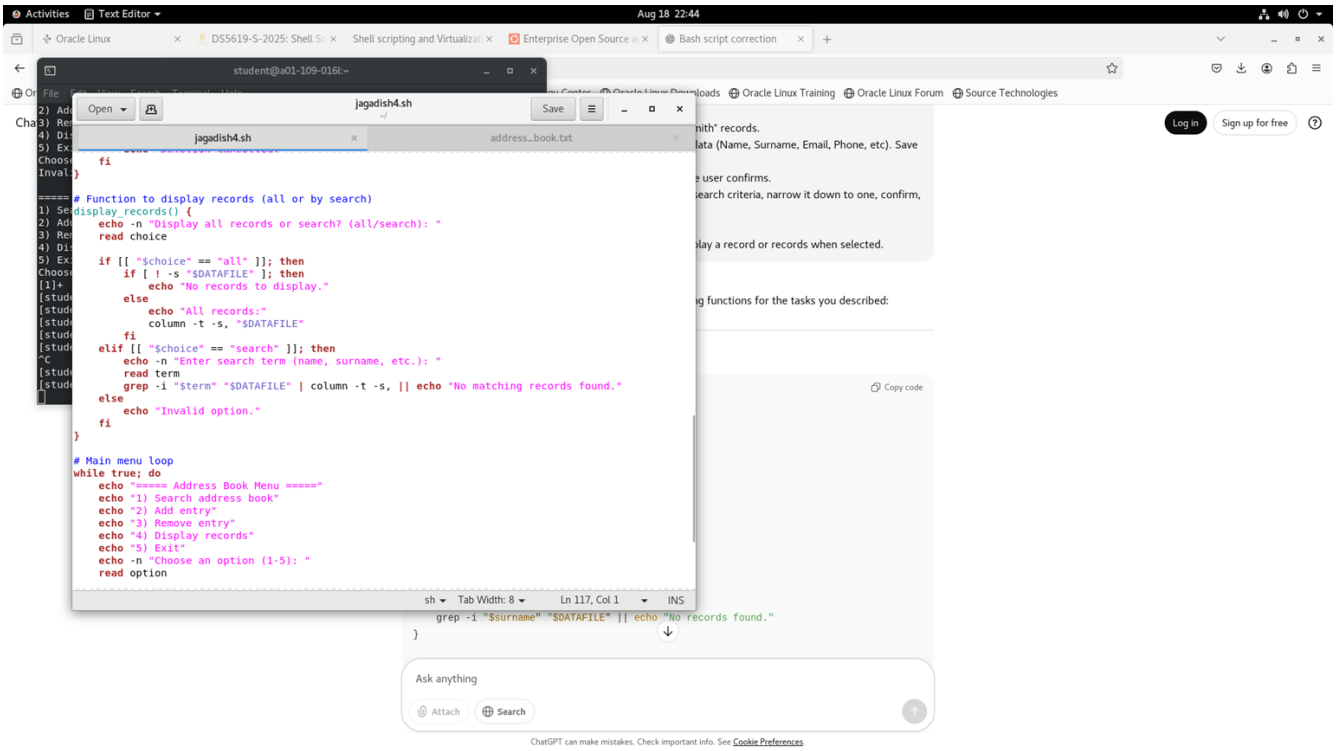
a. Search address book: When the user searches for "Smith", the script should identify and display all "Smith" records.

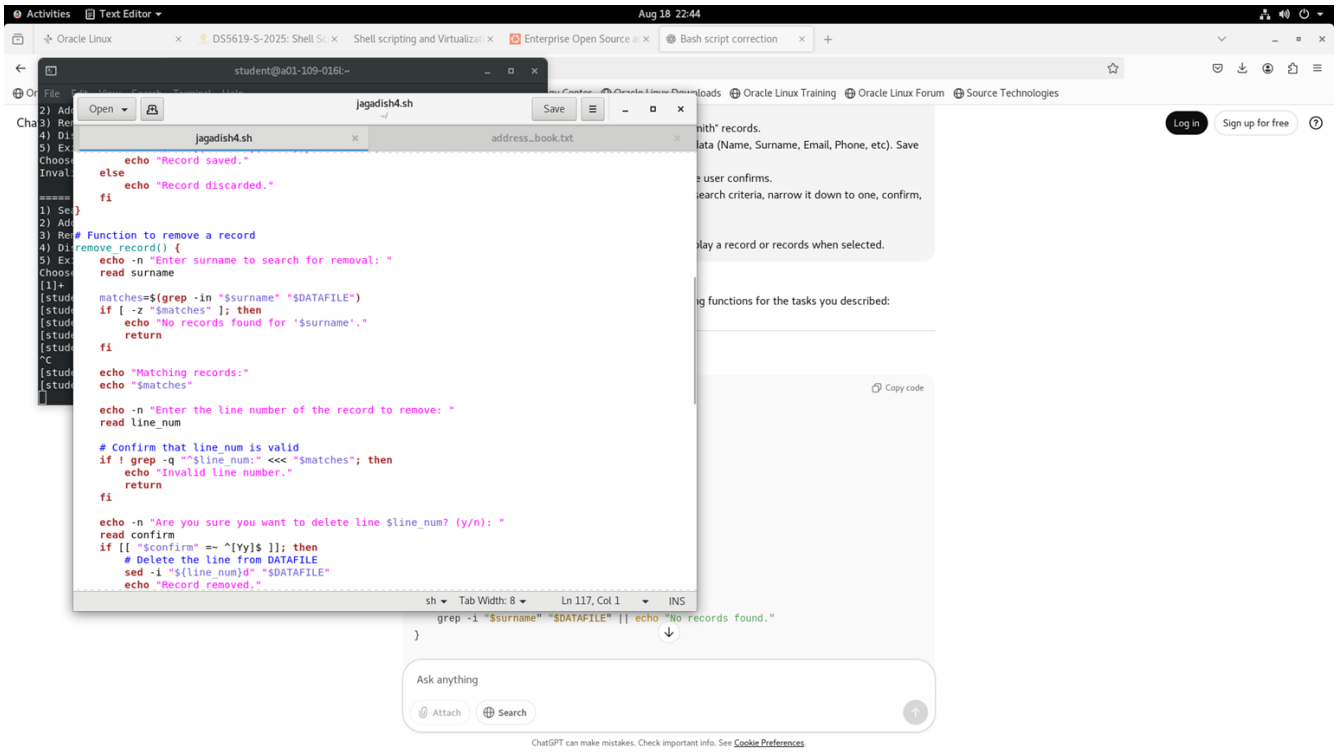
b. Add entries: Input the data (Name, Surname, Email, Phone, etc). Save the record into the data file when the user confirms.

c. Remove entries: Enter search criteria, narrow it down to one, confirm, and then remove that record.

d. Display function to display a record or records when selected.




Activities
Text Editor
Aug 18 22:44
Oracle Linux
DS5619-S-2025: Shell Sc
Shell scripting and Virtualizat
Enterprise Open Source a
Bash script correction
student@a01-109-016l:~
Open
jagadish4.sh
~/
Save
jagadish4.sh
address_book.txt
fi
}
# Function to display records (all or by search)
display_records() {
echo -n "Display all records or search? (all/search): "
read choice
if [["$choice" == "all"]]; then
if [! -s "$DATAFILE"]; then
echo "No records to display."
else
echo "All records:"
column -t -s, "$DATAFILE"
fi
elif [["$choice" == "search"]]; then
echo -n "Enter search term (name, surname, etc.): "
read term
grep -i "$term" "$DATAFILE" | column -t -s, || echo "No matching records found."
else
echo "Invalid option."
fi
}
# Main menu loop
while true; do
echo "===== Address Book Menu ====="
echo "1) Search address book"
echo "2) Add entry"
echo "3) Remove entry"
echo "4) Display records"
echo "5) Exit"
echo -n "Choose an option (1-5): "
read option
sh
Tab Width: 8
Ln 117, Col 1
INS
Oracle Linux Training
Oracle Linux Forum
Source Technologies
Log in
Sign up for free
Copy code
grep -i "$surname" "$DATAFILE" || echo "No records found."
}
Ask anything
Attach
Search
ChatGPT can make mistakes. Check important info. See Cookie Preferences.


Activities
Text Editor
Aug 18 22:44
Oracle Linux
DS5619-S-2025: Shell Sc
Shell scripting and Virtualizat
Enterprise Open Source a
Bash script correction
student@a01-109-016l:~
Open
jagadish4.sh
~/
Save
jagadish4.sh
address_book.txt
echo "Record saved."
else
echo "Record discarded."
fi
}
# Function to remove a record
remove_record() {
echo -n "Enter surname to search for removal: "
read surname
matches=$(grep -in "$surname" "$DATAFILE")
if [-z "$matches"]; then
echo "No records found for '$surname'."
return
fi
echo "Matching records:"
echo "$matches"
echo -n "Enter the line number of the record to remove: "
read line_num
# Confirm that line_num is valid
if ! grep -q "^$line_num:" <<< "$matches"; then
echo "Invalid line number."
return
fi
echo -n "Are you sure you want to delete line $line_num? (y/n): "
read confirm
if [["$confirm" =~ ^[Yy]$]]; then
# Delete the line from DATAFILE
sed -i "${line_num}d" "$DATAFILE"
echo "Record removed."
sh
Tab Width: 8
Ln 117, Col 1
INS
Oracle Linux Training
Oracle Linux Forum
Source Technologies
Log in
Sign up for free
Copy code
grep -i "$surname" "$DATAFILE" || echo "No records found."
}
Ask anything
Attach
Search
ChatGPT can make mistakes. Check important info. See Cookie Preferences.

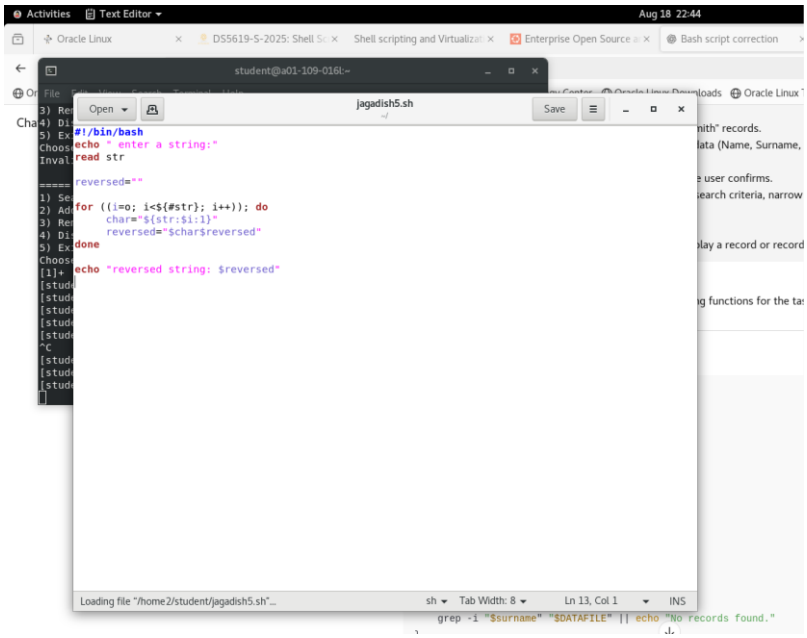
```bash
#!/bin/bash

DATAFILE="address_book.txt"

# Make sure data file exists
touch "$DATAFILE"

# Function to search records by surname
search_records() {
    echo -n "Enter surname to search: "
    read surname
    echo "Search results for '$surname':"
    grep -i "$surname" "$DATAFILE" || echo "No records found."
}

# Function to add a new record
add_record() {
    echo "Enter details to add a new record."
    read -p "Name: " name
    read -p "Surname: " surname
    read -p "Email: " email
    read -p "Phone: " phone

    echo "You entered:"
    echo "Name: $name"
    echo "Surname: $surname"
    echo "Email: $email"
    echo "Phone: $phone"
    echo -n "Save this record? (y/n): "
    read confirm

    if [[ "$confirm" =~ ^[Yy]$ ]]; then
        echo "$name,$surname,$email,$phone" >> "$DATAFILE"
        echo "Record saved."
    else
        echo "Record discarded."
    fi
}

# Function to remove a record
remove_record() {
    echo -n "Enter surname to search for removal: "
    read surname

    matches=$(grep -in "$surname" "$DATAFILE")
    if [ -z "$matches" ]; then
        echo "No records found for '$surname'."
        return
    fi
```

```
    echo "Matching records:"
    echo "$matches"

    echo -n "Enter the line number of the record to remove: "
    read line_num

    # Confirm #!/bin/bash

DATAFILE="address_book.txt"

# Make sure data file exists
touch "$DATAFILE"

# Function to search records by surname
search_records() {
    echo -n "Enter surname to search: "
    read surname
    echo "Search results for '$surname':"
    grep -i "$surname" "$DATAFILE" || echo "No records found."
}

# Function to add a new record
add_record() {
    echo "Enter details to add a new record."
    read -p "Name: " name
    read -p "Surname: " surname
    read -p "Email: " email
    read -p "Phone: " phone

    echo "You entered:"
    echo "Name: $name"
    echo "Surname: $surname"
    echo "Email: $email"
    echo "Phone: $phone"
    echo -n "Save this record? (y/n): "
    read confirm

    if [[ "$confirm" =~ ^[Yy]$ ]]; then
        echo "$name,$surname,$email,$phone" >> "$DATAFILE"
        echo "Record saved."
    else
        echo "Record discarded."
    fi
}

# Function to remove a record
remove_record() {
    echo -n "Enter surname to search for removal: "
```

```
    read surname

    matches=$(grep -in "$surname" "$DATAFILE")
    if [ -z "$matches" ]; then
        echo "No records found for '$surname'."
        return
    fi

    echo "Matching records:"
    echo "$matches"

    echo -n "Enter the line number of the record to remove: "
    read line_num

    # Confirm that line_num is valid
    if ! grep -q "^$line_num:" <<< "$matches"; then
        echo "Invalid line number."
        return
    fi

    echo -n "Are you sure you want to delete line $line_num? (y/n): "
    read confirm
    if [[ "$confirm" =~ ^[Yy]$ ]]; then
        # Delete the line from DATAFILE
        sed -i "${line_num}d" "$DATAFILE"
        echo "Record removed."
    else
        echo "Deletion cancelled."
    fi
}

# Function to display records (all or by search)
display_records() {
    echo -n "Display all records or search? (all/search): "
    read choice

    if [[ "$choice" == "all" ]]; then
        if [ ! -s "$DATAFILE" ]; then
            echo "No records to display."
        else
            echo "All records:"
            column -t -s, "$DATAFILE"
        fi
    elif [[ "$choice" == "search" ]]; then
        echo -n "Enter search term (name, surname, etc.): "
        read term
        grep -i "$term" "$DATAFILE" | column -t -s, || echo "No matching records found."
    else
        echo "Invalid option."
```

```
    fi
}

# Main menu loop
while true; do
    echo "===== Address Book Menu ======"
    echo "1) Search address book"
    echo "2) Add entry"
    echo "3) Remove entry"
    echo "4) Display records"
    echo "5) Exit"
    echo -n "Choose an option (1-5): "
    read option

    case $option in
        1) search_records ;;
        2) add_record ;;
        3) remove_record ;;
        4) display_records ;;
        5) echo "Goodbye!"; exit 0 ;;
        *) echo "Invalid option, try again." ;;
```

5.Write a shell script to reverse a string




```
#!/bin/bash
echo " enter a string:"
read str
```

```
reversed=""

for ((i=o; i<${#str}; i++)); do
    char="${str:$i:1}"
    reversed="$char$reversed"
done

echo "reversed string: $reversed"
```