# Calculator

#!/bin/bash

# Prompt the user for input

echo "Enter an operation (add, subtract, multiply, divide): " read operation

# Prompt the user for numbers echo "Enter first number: "

read num1

echo "Enter second number: "

read num2

# Perform the operation based on user input case $operation in

"add")

result=$((num1 + num2))

;;

"subtract")

result=$((num1 - num2))

;;

"multiply")

result=$((num1 \* num2))

;;

"divide")

result=$((num1 / num2))

;;

\*)

esac

echo "Invalid operation" exit 1

;;

# Display the result

echo "Result: $result"

# Reverse String

#!/bin/bash

echo "Enter a string: " read input\_string

len=${#input\_string}

reversed=""

for (( i=$len-1; i>=0; i-- )); do

reversed="$reversed${input\_string:$i:1}" done

echo "Reversed string: $reversed"

# Menu Driven Program Using Swich Case

#!/bin/bash

# Function to display current date display\_date() {

echo "Current date: $(date)"

}

# Function to display current user display\_user() {

echo "Current user: $(whoami)"

}

# Function to display total number of files in current directory display\_files\_count() {

local count=$(ls -1 | wc -l)

echo "Total number of files in current directory: $count"

}

# Main menu while true; do

echo "Menu:"

echo "1. Display current date" echo "2. Display current user"

echo "3. Display total number of files in current directory" echo "4. Exit"

echo "Enter your choice: " read choice

case $choice in 1)

display\_date

;;

2)

display\_user

;;

3)

4)

\*)

esac done

display\_files\_count

;;

echo "Exiting..." break

;;

echo "Invalid choice. Please try again."

;;

# Print Pyramid

#!/bin/bash

# Prompt user to enter the number of rows echo "Please enter the number of rows:" read num\_rows

# Loop through the rows

for (( i=1; i<=num\_rows; i++ )); do # Print spaces for the current row

for (( j=1; j<=num\_rows-i; j++ )); do echo -n " "

done

# Print asterisks for the current row for (( j=1; j<=2\*i-1; j++ )); do

echo -n "\*" done

# Move to the next line echo ""

done

# Factorial

#!/bin/bash

echo "Enter a number:" read num

fact=1

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

fact=$((fact \* i)) done

echo "The factorial of $num is $fact"

* 1. **SORTING Bubble Sort** #!/bin/bash

# Function to swap two numbers swap() {

local temp="${arr[$1]}" arr[$1]="${arr[$2]}" arr[$2]="$temp"

}

# Function for bubble sort bubble\_sort() {

local n="${#arr[@]}"

for ((i=0; i<$n-1; i++)); do

for ((j=0; j<$n-$i-1; j++)); do

if ((arr[j] > arr[j+1])); then swap $j $((j+1))

fi done

done

}

# Main script

echo "Enter the number of elements: " read num

echo "Enter the elements separated by space: "

read -a arr

echo "Original array: ${arr[@]}"

bubble\_sort

echo "Sorted array: ${arr[@]}"

**QuickSort**

#!/bin/bash

# Function to perform quicksort quicksort() {

local arr=("$@") # Convert command line arguments into an array local left=()

local right=() local pivot

if (( ${#arr[@]} <= 1 )); then

echo "${arr[@]}" # Return the sorted array return

fi

pivot="${arr[0]}" # Choose the first element as pivot for num in "${arr[@]:1}"; do

if (( num < pivot )); then

left+=("$num") # Append to the left array if smaller

else

right+=("$num") # Append to the right array if larger

fi done

# Recursively sort left and right arrays, then combine with pivot echo "$(quicksort "${left[@]}") $pivot $(quicksort "${right[@]}")"

}

# Main script

echo "Enter the number of elements: " read num

echo "Enter the elements separated by space: " read -a arr

echo "Original array: ${arr[@]}"

sorted\_arr=($(quicksort "${arr[@]}")) # Call the quicksort function and store the sorted array

echo "Sorted array: ${sorted\_arr[@]}"

# Binary To Decimal

#!/bin/bash

echo "Enter a binary number: " read binary

decimal=0 i=0

while [ $binary -ne 0 ] do

digit=$(( binary % 10 ))

decimal=$(( decimal + digit \* 2\*\*i )) binary=$(( binary / 10 ))

i=$(( i + 1 )) done

echo "Decimal number is: $decimal"