* 1. **Write a shell script which accepts a number and displays the list of odd numbers below that number. It should also display the sum of all this odd numbers.**

echo "Enter a number: "

read n sum=0

echo "List of odd numbers below $n:"

for (( i=1; i<n; i+=2 ))

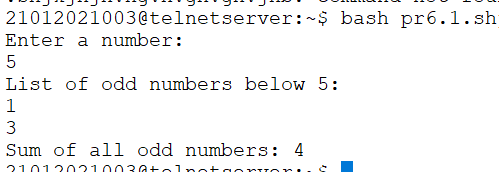
do

echo $i

sum=$((sum+i))

done

echo "Sum of all odd numbers: $sum"



* 1. **Write a shell script to arrange numbers in ascending or descending order as per the user choice.**

echo "Enter the number of elements: "

read n

echo "Enter the elements: "

for (( i=0; i<n; i++ ))

do

read a[$i]

done

echo "Enter 1 to sort in ascending order or 2 to sort in descending order: "

read choice if [ $choice -eq 1 ]

then

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

do

for (( j=i+1; j<n; j++ ))

do

if [ ${a[i]} -gt ${a[j]} ]

then

temp=${a[i]} a[i]=${a[j]} a[j]=$temp

fi

done

done

echo "Elements in ascending order: ${a[@]}"

elif [ $choice -eq 2 ]

then

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

do

for (( j=i+1; j<n; j++ ))

do

if [ ${a[i]} -lt ${a[j]} ]

then

temp=${a[i]} a[i]=${a[j]} a[j]=$temp

fi

done

done

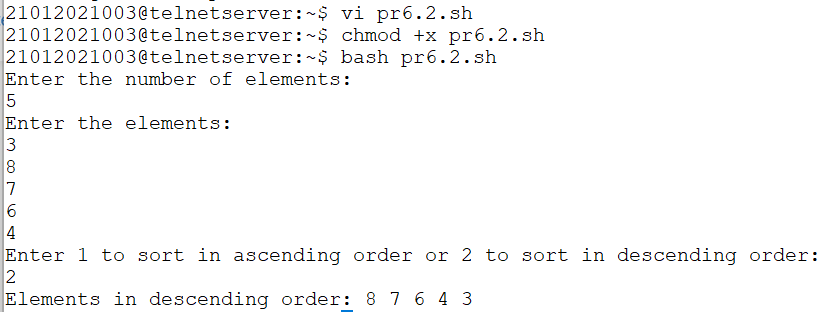
echo "Elements in descending order: ${a[@]}"

else

echo "Invalid choice."

fi

1. Bottom of Form



1. **Write a shell script to check whether the entered number is Armstrong or not.**

echo "Enter a number:"

read num

num\_of\_digits=${#num}

sum=0

for (( i=0; i<$num\_of\_digits; i++ )); do

digit=${num:$i:1}

(( sum += $digit \*\* $num\_of\_digits ))

done

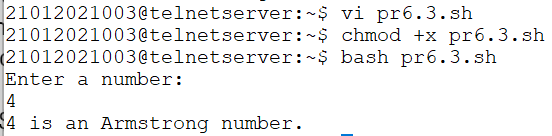
if [[ $sum -eq $num ]]; then

echo "$num is an Armstrong number."

else

echo "$num is not an Armstrong number."

fi

****

1. **Size of array A is 10 while size of B is 30. Scan 10 integers in both the array and concat array A to B. Then apply sorting algorithm according to the user choice.**

# initialize arrays

A=()

B=()

# prompt user to enter values for array A

echo "Enter 10 integers for array A:"

for (( i=0; i<10; i++ )); do

read num

A+=("$num")

done

# prompt user to enter values for array B

echo "Enter 30 integers for array B:"

for (( i=0; i<30; i++ )); do

read num

B+=("$num")

done

# concatenate arrays A and B

C=( "${A[@]}" "${B[@]}" )

# prompt user to choose sorting algorithm

echo "Choose a sorting algorithm (1 for bubble sort, 2 for selection sort, 3 for insertion sort):"

read choice

# sort the array according to user choice

if [[ $choice -eq 1 ]]; then

# bubble sort

for (( i=0; i<${#C[@]}-1; i++ )); do

for (( j=0; j<${#C[@]}-1-$i; j++ )); do

if [[ ${C[j]} -gt ${C[j+1]} ]]; then

# swap elements

temp=${C[j]}

C[j]=${C[j+1]}

C[j+1]=$temp

fi

done

done

elif [[ $choice -eq 2 ]]; then

# selection sort

for (( i=0; i<${#C[@]}-1; i++ )); do

min=$i

for (( j=$i+1; j<${#C[@]}; j++ )); do

if [[ ${C[j]} -lt ${C[min]} ]]; then

min=$j

fi

done

# swap elements

temp=${C[i]}

C[i]=${C[min]}

C[min]=$temp

done

elif [[ $choice -eq 3 ]]; then

# insertion sort

for (( i=1; i<${#C[@]}; i++ )); do

key=${C[i]}

j=$i-1

while [[ $j -ge 0 && ${C[j]} -gt $key ]]; do

# shift elements

C[$j+1]=${C[$j]}

(( j-- ))

done

C[$j+1]=$key

done

else

echo "Invalid choice."

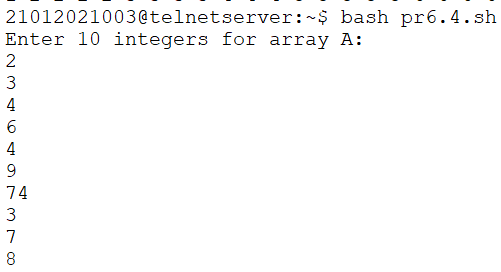
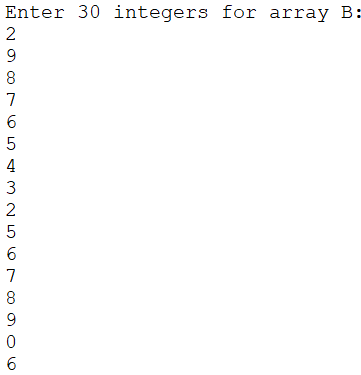
exit 1

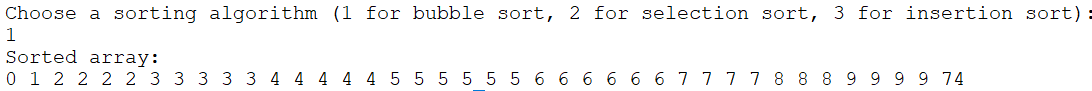
fi

# print the sorted array

echo "Sorted array:"

echo "${C[@]}"



1. **Write a shell script to remove duplicates values from an array.**

arr=(1 2 3 3 4 5 5 6)

for i in "${!arr[@]}"; do

for j in "${!arr[@]}"; do

if [[ "${arr[$i]}" = "${arr[$j]}" && "$i" -ne "$j" ]]; then

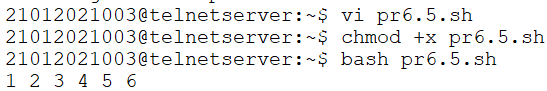
unset arr[$j]

fi

done

done

echo "${arr[@]}"



1. **Write a shell script to add two arrays.**

A=(1 2 3 4 5)

B=(6 7 8 9 10)

len\_A=${#A[@]}

len\_B=${#B[@]}

if [ $len\_A -ne $len\_B ]

then

echo "Error: Arrays A and B have different lengths"

exit 1

fi

C=()

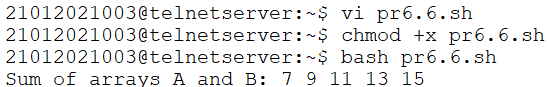
for i in $(seq 0 $(($len\_A-1)))

do

C+=( $(( ${A[$i]} + ${B[$i]} )) )

done

echo "Sum of arrays A and B: ${C[@]}"



1. **Write a shell script to reverse an array.**

arr=(1 2 3 4 5)

len=${#arr[@]}

for (( i=0; i<$len/2; i++ )); do

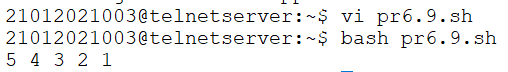
temp=${arr[i]}

arr[i]=${arr[$len-i-1]}

arr[$len-i-1]=$temp

done

echo "${arr[@]}"



1. **Write a shell script to check whether the entered string is in title case or not.**

echo "Enter a string: "

read string

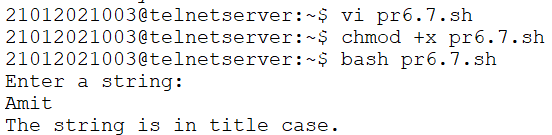
if [[ $string =~ ^[[:upper:]][[:lower:]]+(\ [[:upper:]][[:lower:]]+)\*$ ]]; then

echo "The string is in title case."

else

echo "The string is not in title case."

fi



1. **Write a shell script to check whether the scanned word is a uppercase word or not.**

echo "Enter a word: "

read word

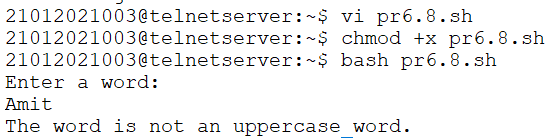
if [[ $word =~ ^[[:upper:]]+$ ]]; then

echo "The word is an uppercase word."

else

echo "The word is not an uppercase word."

fi



1. **Write a shell script to count number of uppercase words in a string.**

echo "Enter a string: "

read string

words=($string)

count=0

for word in "${words[@]}"; do

if [[ $word =~ ^[[:upper:]]+$ ]]; then

((count++))

fi

done

echo "The string contains $count uppercase words."

