# Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology, Pune-37

*(Anautonomous Institute of Savitribai Phule Pune University)*



**Department of Multidisciplinary Engineering**

|  |  |
| --- | --- |
| **Division** | **CS-A** |
| **Batch** | **B1** |
| **Roll no.** | **90** |
| **Name** | **Aditya Shrinivas Kurapati** |

1. Finding The Difference Between Today And Given Date From Input In Days

*#!/bin/bash*

date\_diff() {

    today=$(date +"%Y-%m-%d")

    read -p "Enter a date (YYYY-mm-dd): " userDate

    if [[ ! $userDate =~ ^[0-9]{4}-[0-9]{2}-[0-9]{2}$ ]]; then

      echo "Enter a valid date (YYYY-mm-dd)"

      exit 1

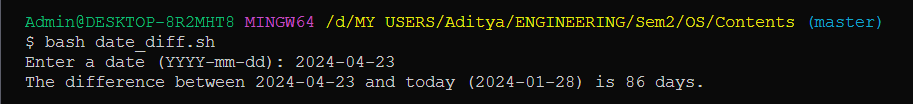
    fi

    dateDiff=$(( ($(date -d "$userDate" +%s) - $(date -d "$today" +%s)) / 86400 ))

    echo "The difference between $userDate and today ($today) is $dateDiff days."

}

date\_diff



Concepts Covered – Function, Condition Statements , Commands, Accepting Input , Arithmetic Operations.

2. Using ‘\*’ Designing ‘L’ Pattern In Shell Script .

*#!/bin/bash*

printStar() {

    size=5

    for ((i = 1; i <= size; i++)); do

        for ((j = 1; j <= size; j++)); do

            if [ $i -eq $((size / 2 + 1)) ] || [ $j -eq $((size / 2 + 1)) ]; then

                echo -n "xx"

            else

                echo -n "  "

            fi

        done

        echo ""

    done

}

printStar



Concepts Covered – Pattern Creation, Loops(Control Structures), Functions , Conditional Statements

3. Finding Unique Elements From Given Array/ Removing Duplicate Elements From Array

*#!bin/bash*

myArray=(*$1* *$2* *$3* *$4* *$5* *$6* *$7* *$8*)

for element in "${myArray[@]}"; do

  if [[ ! " ${uniqueArray[@]} " =~ "$element" ]]; then

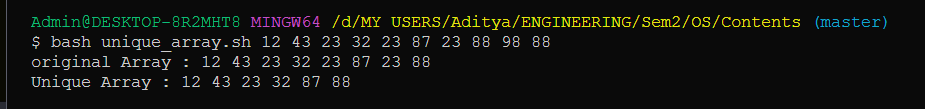
    uniqueArray+=("$element")

  fi

done

echo "original Array : ${myArray[@]}"

echo "Unique Array : ${uniqueArray[@]}"



Concepts Covered – Command Line Arguments , Arrays ,Strings (Dynamic Strings),Conditional Statements , Control Structures

4. Shell Script To Manipulate Strings

*#!/bin/bash*

reverseString() {

    read -p "Enter a string: " inputString

    reversedString=""

    length=${#inputString}

    for ((i = length - 1; i >= 0; i--)); do

        reversedString="${reversedString}${inputString:$i:1}"

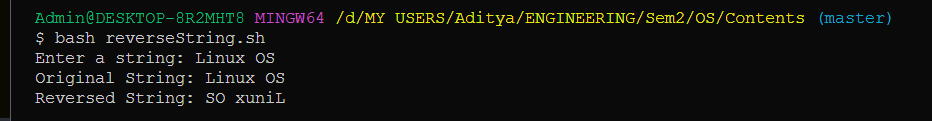
    done

    echo "Original String: $inputString"

    echo "Reversed String: $reversedString"

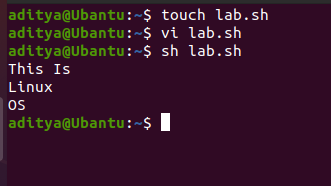
}

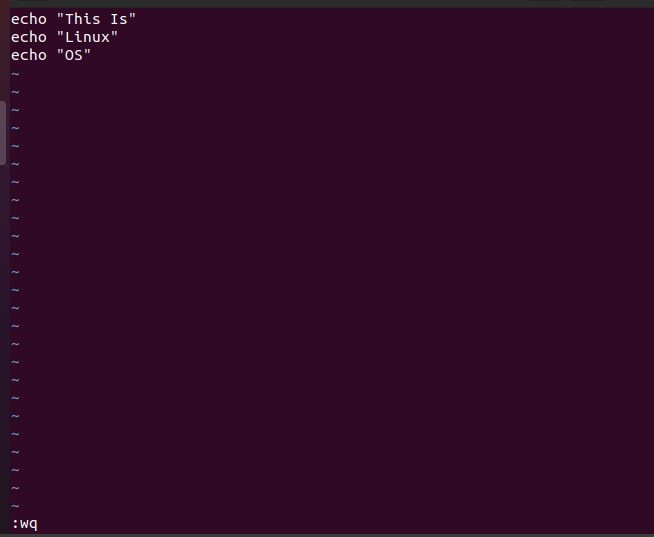
reverseString



Concepts Covered – String Operation, Control Structures ,Functions

5. Handling File Operations





Concepts Covered – File Creation And handling Using Vi Editor