

Assignment 4

Software Tools and Technologies

Name : Ravish Ranjan
Course : MCA
Semester : 1st semester

Write and submit the following shell scripts (.sh files). Each script must be properly commented and tested.

Questions

1. System Information Script

Display:
- Current date and time
- Logged-in username
- System uptime
- Disk usage of root (/) partition
- Save output in a file named sysinfo.txt.

```
#!/bin/bash

outputfile="sysinfo.txt"

{
    echo -n -e "\e[1;34mDate & Time      : \e[0m"
    date +"%A, %d %B %Y %I:%M:%S %p" # displaying date and time in formated manner

    echo -n -e "\e[1;34mLogged-in User : \e[0m"
    whoami # displaying username

    echo -n -e "\e[1;34mSystem uptime   : \e[0m"
    uptime -p # displaying uptime of system in formatted manner

    echo -n -e "\e[1;34mDisk usage (/) : \e[0m"
    df -h / | awk 'NR==2 {print "Used : " $3 "(" $5 ") | Available : " $4 " |"
    Total : " $2 " | Mount point : " $6}' # displaying disusage in formatted manner
} > "$outputfile"

echo "System information saved in file $outputfile"
```

```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./sysinfo.sh
System information saved in file
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ls sysinfo*
sysinfo.sh  sysinfo.txt
```

Output

```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ cat sysinfo.txt
Date & Time      : Friday, 17 October 2025 06:00:14 PM
Logged-in User   : ravish
System uptime    : up 1 hour, 22 minutes
Disk usage (/)   : Used : 1.9G (1%) | Available : 954G | Total : 1007G | Mount point : /
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ |
```

2. Simple Calculator Script

- Accept two numbers from the user.
- Provide a menu with four choices: Addition, Subtraction, Multiplication, Division.
- Perform the selected operation and display the result.

```
#!/bin/bash

firstnum=$1 # getting first number from file call args
operation=$2 # getting operation from file call args
secondnum=$3 # getting second number from file call args

# getting first number if not provided
if [ -z "$firstnum" ]; then
    read -p "Enter first number : " firstnum
fi

# validating first input
if ! [[ $firstnum =~ ^[0-9]+$ ]]; then
    echo -e "\e[1;31mError:\e[0m Not a valid number" >&2 # printing error
    exit 2 # exiting with error code 2
fi

# getting operation to perform if not provided
if [ -z "$operation" ]; then
    read -p "Enter operation (+ - * /) : " operation
fi
operation=$(echo "$operation" | tr -d '\r' | xargs)

# valudating operation
if ! [[ "$operation" == "+" || "$operation" == "-" || "$operation" == "*" || "$operation" == "/" ]]; then
    echo -e "\e[1;31mError:\e[0m Not a valid operation" >&2 # printing error
    exit 2 # exiting with error code 2
fi
```

```

# getting second number if not provided
if [ -z "$secondnum" ]; then
    read -p "Enter second number : " secondnum
fi

# validating second input
if ! [[ $secondnum =~ ^[0-9]+$ ]]; then
    echo -e "\e[1;31mError:\e[0m Not a valid number" >&2 # printing error
    exit 2 # exiting with error code 2
fi

# catching if divide by zero opeation is given
if [[ "$operation" == "/" && "$secondnum" -eq 0 ]]; then
    echo -e "\e[1;31mError:\e[0m Division by zero" >&2
    exit 2
fi

res=$(( firstnum $operation secondnum )) # calculating result
echo -e "\e[1;34mResult :\e[0m $res" # displaying result

```

Output

```

ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./calc.sh 8 + 9
Result : 17
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./calc.sh
Enter first number : 34
Enter operation (+ - * /) : *
Enter second number : 43
Result : 1462
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./calc.sh 50
Enter operation (+ - * /) : /
Enter second number : 10
Result : 5
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./calc.sh 80 -
Enter second number : 20
Result : 60
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./calc.sh
Enter first number : a
Error: Not a valid number
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./calc.sh
Enter first number : 2
Enter operation (+ - * /) : &
Error: Not a valid operation
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./calc.sh
Enter first number : 2
Enter operation (+ - * /) : +
Enter second number : r
Error: Not a valid number
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ |

```

3. Odd/Even Checker

- Accept a number from the user.
- Use if conditions to check whether it is odd or even and print the result.

```
#!/bin/bash

number=$1

if [ -z "$number" ]; then
    read -p "Enter a number : " number
fi

if ! [[ "$number" =~ ^[0-9]+$ ]]; then
    echo -e "\e[1;31mError:\e[0m Not a valid number" >&2
    exit 2
fi

if (( $number % 2 == 0 )); then
    echo -e "$number is \e[1;34mEven"
else
    echo -e "$number is \e[1;34mOdd"
fi
```

Output

```
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./oddeven.sh
Enter a number : 2342
2342 is Even
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ ./oddeven.sh 453
453 is Odd
ravish@RavishPC:/mnt/c/work/sem1/stt/prog$ |
```