NORTHERN UNIVERSITY

Knowledge for Innovation and Change

COURSE TITLE: OPERATING SYSTEM LAB WORK
COURSE CODE: CSE 3373

REPORT ON: IMPLEMENTATION OF CONDITIONAL STATEMENTS IN BASH

SUBMITTED TO,

NIZIA NAHYAN
LECTURER, NUB(CSE)

SUBMITTED BY,

ZUBAER AHMED ZIHAD
ID: 41230100877
SECTION: 7A

DATE OF SUBMISSION: 04 May 2025

Lab Report: Implementation of Conditional Statements in Bash

Introduction:

Bash scripting allows users to automate tasks in Unix/Linux environments. One of the key capabilities of Bash is decision-making using conditional statements. These allow scripts to make logical decisions, execute specific code blocks, and respond dynamically to user input. This lab focuses on the implementation of conditional and nested conditional statements using if, else, and relational operators like -ge and -le.

Code Example 1: ID Range Validation Using Nested If-Else

Bash

```bash
echo "Enter your ID : "
read id
if [ $id -ge 1 ]
then
  if [ $id -le 200 ]
  then
    echo "Your ID is is between 1 - 199"
  else
    echo "Your ID is grater than 200"
  fi
else
  echo "ID is less than 1"
fi
  echo "Zobaer Ahmed Zihad ID: 877"
```

Output: 01

```
0
```

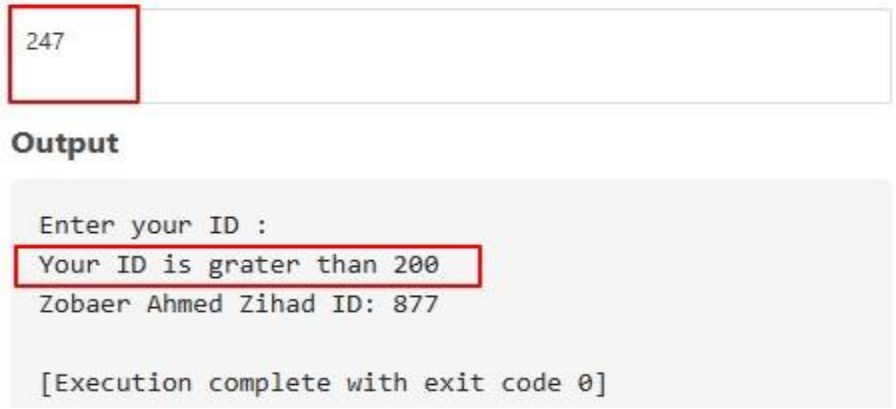
Output

```
Enter your ID :
ID is less than 1
Zobaer Ahmed Zihad ID: 877

[Execution complete with exit code 0]
```

Output: 02

Output: 03



Explanation:

This nested if-else block first checks if the entered ID is greater than or equal to 1. If true, it then checks if the ID is less than or equal to 200. If both conditions are met, the ID is considered valid. Otherwise, it is flagged as invalid.

Code Example 2: Find the Greatest Among Three Numbers Using Nested If-Else.

Bash Code:

```bash
read a
read b
read c
if [ $a -ge $b ]
    then
    if [ $a -ge $c ]
    then
        echo "a is greater."
    else
        echo "c is greater."
    fi
else
    if [ $b -ge $c ]
    then
        echo "b is greater."
    else
        echo "c is greater."
    fi
fi
    echo "Zobaer Ahmed Zihad ID: 877"
```

Output: 01

Output: 02

Output: 03

Explanation: This nested conditional structure compares three numbers and determines the greatest among them. The logic is divided into multiple branches to systematically evaluate all comparisons.

Discussion: Conditional and nested conditional statements are fundamental in Bash for building logical decision-making paths. Using operators like -ge and - le within if blocks allows scripts to validate inputs and compare numerical values effectively. Nested if-else structures help handle more complex logic, such as range validation and multi-variable comparisons. This lab demonstrates how such constructs are used in practice to build responsive and robust Bash scripts.