Antony mawira assignment work

#!/bin/bash

# Prompt user to enter employee details

read -p "Enter employee name: " name

read -p "Enter hours worked: " hours

read -p "Enter rate per hour: " rate

# Calculate basic pay

basic\_pay=$((hours \* rate))

# Calculate tax based on basic pay

if [ $basic\_pay -gt 70000 ]; then

tax=$((basic\_pay \* 25 / 100))

elif [ $basic\_pay -ge 15000 ]; then

tax=$((basic\_pay \* 15 / 100))

else

tax=0

fi

# Calculate net pay

net\_pay=$((basic\_pay - tax))

# Display results

echo "Employee Name: $name"

echo "Basic Pay: $basic\_pay"

echo "Tax: $tax"

echo "Net Pay: $net\_pay"

2

#!/bin/bash

# Prompt user to enter CustomerID, CustomerName, and UnitConsumed

read -p "Enter CustomerID: " customerID

read -p "Enter CustomerName: " customerName

read -p "Enter UnitConsumed: " unitConsumed

# Calculate total bill based on the charging rates

if [ $unitConsumed -lt 200 ]; then

totalBill=$(( $unitConsumed \* 120 ))

elif [ $unitConsumed -ge 200 ] && [ $unitConsumed -lt 400 ]; then

totalBill=$(( $unitConsumed \* 150 ))

elif [ $unitConsumed -ge 400 ] && [ $unitConsumed -lt 600 ]; then

totalBill=$(( $unitConsumed \* 180 ))

else

totalBill=$(( $unitConsumed \* 200 ))

fi

echo "CustomerID: $customerID"

echo "CustomerName: $customerName"

echo "Total Bill: Ksh $totalBill"

3

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <fcntl.h>

#include <unistd.h>

int main() {

int fd; // file descriptor

char buf[100]; // buffer to read file content

ssize\_t nread; // number of bytes read

// open file for writing

fd = open("hello.txt", O\_CREAT | O\_WRONLY, S\_IRUSR | S\_IWUSR);

if (fd == -1) {

perror("open");

exit(EXIT\_FAILURE);

}

// write "Hello World" to file

if (write(fd, "Hello World", strlen("Hello World")) == -1) {

perror("write");

exit(EXIT\_FAILURE);

}

// close file

if (close(fd) == -1) {

perror("close");

exit(EXIT\_FAILURE);

}

// open file for reading

fd = open("hello.txt", O\_RDONLY);

if (fd == -1) {

perror("open");

exit(EXIT\_FAILURE);

}

// read file content into buffer

nread = read(fd, buf, sizeof(buf));

if (nread == -1) {

perror("read");

exit(EXIT\_FAILURE);

}

// print read content

printf("Read from file: %.\*s\n", (int)nread, buf);

// close file

if (close(fd) == -1) {

perror("close");

exit(EXIT\_FAILURE);

}

return 0;

}