PF Lab 02

Task 1:

Pseudo Code:

```
1 START
2 DISPLAY "Here is the menu"
3 READ customer_order
4 IF order is not available THEN
5       DISPLAY "Order is not available"
6       END
7 DISPLAY "Any special requests?"
8 READ special_requests
9 IF special_requests is not empty THEN
10      READ special_request_price
11      INCREASE amount by special_request_price
12 END IF
13 DISPLAY "Enter total amount for the order:"
14 READ amount
15 DISPLAY "Enter payment amount:"
16 READ customer_payment
17 CALCU change = customer_payment - amount
18 IF customer_payment < amount THEN
19      DISPLAY "Your payment is not sufficient"
20      END
21 DISPLAY "Your order is confirmed. Total: ", amount
```

```
22 IF customer_payment > amount THEN
23      DISPLAY "Here is the change: ", change
24      DISPLAY "Thank you, please wait for your order"
25 END
```

ALGORITHM:

START

Display menu to the customer

Ask the customer their order

Check if the order is available

If the order is not available, inform the customer and ask for a different order

Ask the customer for any special requests

If there are special requests, add the additional cost to the total

Calculate the total cost, including the order and any special requests

Display the total cost to the customer

Ask the customer for the payment

If payment is less than the total cost, ask for additional payment

Set the change to (payment – total cost)

Display the change to the customer

Display the waiting time for the customer

END

TASK 2:

Pseudo Code:

```
1 START
2 DISPLAY "Insert your card"
3 INPUT card
4 IF card is valid THEN
5       DISPLAY "Enter your PIN"
6       INPUT password
7       IF password == PIN THEN
8               DISPLAY "Enter the amount you want to withdraw"
9               INPUT withdrawal_amount
10              IF withdrawal_amount <= bank_balance THEN
11                      Update bank_balance (bank_balance - withdrawal_amount)
12                      DISPLAY "Here is your cash:", withdrawal_amount
13                      DISPLAY "Your remaining bank balance is:", new balance
14              ELSE  DISPLAY "Your balance is insufficient"
15              END IF
16      ELSE  DISPLAY "The entered PIN is incorrect"
17      END IF
18 ELSE  DISPLAY "Invalid card, please try again"
19 END
```

ALGORITHM:

Ask the user to insert card

Check if the card is valid

If the card is invalid, ask the user to try again

If valid, ask the user to enter the PIN

If the PIN is incorrect, ask the user to enter the correct PIN

If the PIN is correct, ask the user to enter the withdrawal amount

Check if sufficient balance exists

If sufficient, give the user the cash and display the updated balance

TASK 3:

Pseudo Code:

```
1 START
2 DISPLAY “Enter three numbers”
3 INPUT num1, num2, num3
4 IF num1 >= num2 AND num1 >= num3 THEN
5       DISPLAY “num1 is the greatest”
6 ELSE IF num2 >= num1 AND num2 >= num3 THEN
7       DISPLAY “num2 is the greatest”
8 ELSE DISPLAY “num3 is the greatest”
9 END
```

ALGORITHM:

Ask the user to enter num1

Ask the user to enter num2

Ask the user to enter num3

If num1 is greater than or equal to num2 and num3 then display num1 is the greatest

If num2 is greater than or equal to num1 and num3 then display num2 is the greatest

If num3 is greater than or equal to num1 and num2 then display num3 is the greatest

TASK 4:

ALGORITHM:

Ask the user to enter a number between 1 and 12

If the number is 1 then display January

If the number is 2 then display February

If the number is 3 then display March

If the number is 4 then display April

If the number is 5 then display May

If the number is 6 then display June

If the number is 7 then display July

If the number is 8 then display August

If the number is 9 then display September

If the number is 10 then display October

If the number is 11 then display November

If the number is 12 then display December

TASK 5:

Pseudo Code:

1 START

2 DISPLAY “Enter the first number”

```
3 NPUT num1
4 DISPLAY “Enter the second number”
5 INPUT num2
6 DISPLAY “Enter the operation (+ or -)”
7 INPUT operator
8 IF operator == “+” THEN
9       result1 = num1 + num2
10      DISPLAY “The result is”, result1
11 ELSE IF operator == “-“ THEN
12              result2 = num2 – num 1
13              DISPLAY “The result is”, result2
14 ELSE DISPLAY “Invalid operator”
15 END
```

TASK 7:

ALGORITHM:

START

Ask the user to enter the first number

Read and store the first number

Ask the user to enter the operator (+, -, *, /)

Read and store the operator

Ask the user to enter the second number

Read and store the second number

If the operator is ‘+’, add the first and second number

If the operator is ‘-‘, subtract the second number from the first number

If the operator is ‘*’, multiply the first and second numbers

If the operator is ‘/’, divide the first number by the second number

Display the result

END

TASK 8:

Why we use .gitignore?

A .gitignore file is used to specify which files and directories should be ignored by version control. When you add files or directories to the .gitignore file, Git will not include them in commits. This is useful because if prevents unnecessary files, temporary files, or system specific files from being tracked. It also protects sensitive informative such as passwords or API keys. It also keeps the repository clean and focuses only on the source code and related files.

TASK 9:

Difference between Algorithm and Psuedo Code?

An algorithm is a step-by-step procedure to solve a specific problem. It is a conceptual idea that outlines a logical sequence needed to achieve a goal, and it is typically describe in a natural language.

On the other hand, pseudocode is an informal, high-level description of an algorithm that uses a combination of natural language and programming like constructs. It serves as a blueprint for implementing the algorithm in code, resembling actual code but without the strict syntax of a programming language. The primary purpose of pseudocode is to bridge the gap between the abstract algorithm and its practical implementation.