Question 1(a)

1. When solving problems on the computer, one of the most difficult problem-solving steps for the problem solver / programmer is writing the **algorithm**.

2. Programmers have two main resources that they need to optimize for cost effectiveness and these resources are **time** and **memory**.

3. The compiler generates **machine** codes.

4. **Semicolon (;)** indicates a null statement in 'C' which doesn't do anything on execution.

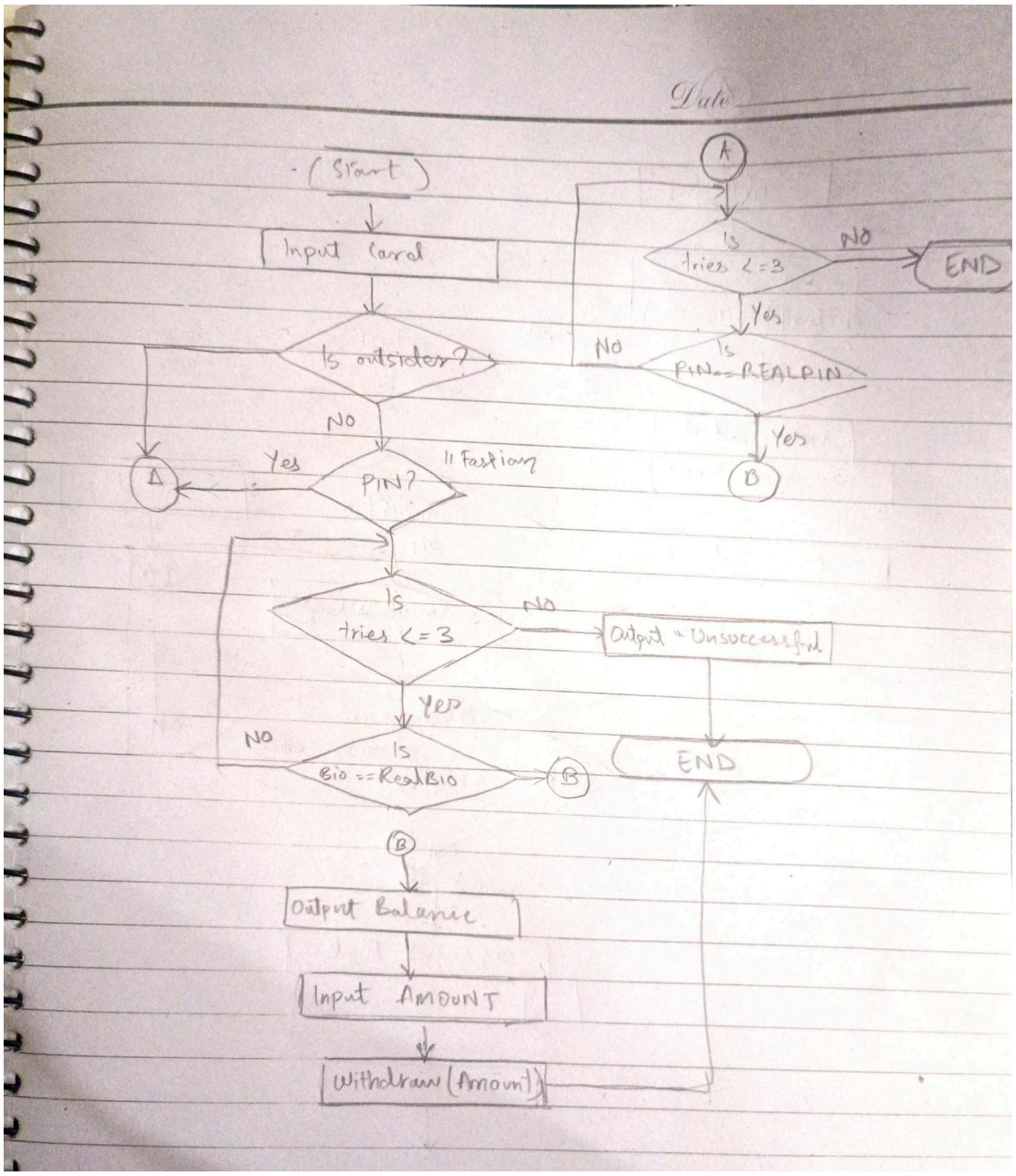
Question 1(b)

1. The program is already correct. It swaps the values of a and b. **Output:** a=5, b=3
2. There is no syntax error in the program, but the logic needs clarification. 5 && 5 evaluates to 1 (true), and 1 == 5 evaluates to false. **Output:** false
3. The division 4/11 is an integer division. Both m and n will get the result of 0 because 4/11 is less than 1 in integer division. **Output:** 0, 0.000000
4. **Correction:** The if (p = q) statement is an assignment, not a comparison. It assigns q to p, and since q is non-zero, the condition is always true. **Output:** Think about it!

Question 2 - PAC

Given data	Required result
ATM Card	Amount
Processing required	Solution alternative
Read card information Separate process for FASTians and Outsiders Validate PIN Perform biometric verification Track and count failed attempts Activate card capture mechanism if failed attempts exceeds the limit of 3 Otherwise Retrieve and display balance Process withdrawal and dispense cash Format and print receipt	Take card data manually instead of using card reader machine

Question 2 - flowchart



Question 3

```
#include <stdio.h>

int main() {
    int accountNumber, basicServiceConnections, premiumChannels;
    char customerType;
    double amountDue = 0.0;

    // Input customer details
    printf("Enter account number (an integer): ");
    scanf("%d", &accountNumber);

    printf("Enter customer type (R or r for Residential, B or b for Business): ");
    scanf(" %c", &customerType);  // Space before %c to ignore any newline characters

    // Calculate bill based on customer type
    if (customerType == 'R' || customerType == 'r') {
        // Residential customer
        double billProcessingFee = 4.50;
        double basicServiceFee = 20.50;
        double premiumChannelFee = 7.50;

        printf("Enter the number of premium channels: ");
        scanf("%d", &premiumChannels);

        // Calculate the total bill for residential customer
        amountDue = billProcessingFee + basicServiceFee + (premiumChannelFee *
premiumChannels);
    }

    else if (customerType == 'B' || customerType == 'b') {
        // Business customer
        double billProcessingFee = 15.00;
        double basicServiceFee = 75.00;
        double additionalConnectionFee = 5.00;
        double premiumChannelFee = 50.00;

        printf("Enter the number of basic service connections: ");
        scanf("%d", &basicServiceConnections);
        printf("Enter the number of premium channels: ");
        scanf("%d", &premiumChannels);

        // Calculate the total bill for business customer
        if (basicServiceConnections > 10) {
```

```
            amountDue = billProcessingFee + basicServiceFee +
                    (basicServiceConnections - 10) * additionalConnectionFee +
                    (premiumChannelFee * premiumChannels);
        } else {
            amountDue = billProcessingFee + basicServiceFee +
                    (premiumChannelFee * premiumChannels);
        }
    }
    else {
        printf("Invalid customer type entered.\n");
        return 1;  // Exit if the customer type is invalid
    }
    // Output the account number and total bill
    printf("Account number: %d\n", accountNumber);
    printf("Amount due: $%.2f\n", amountDue);
    return 0;
}
```