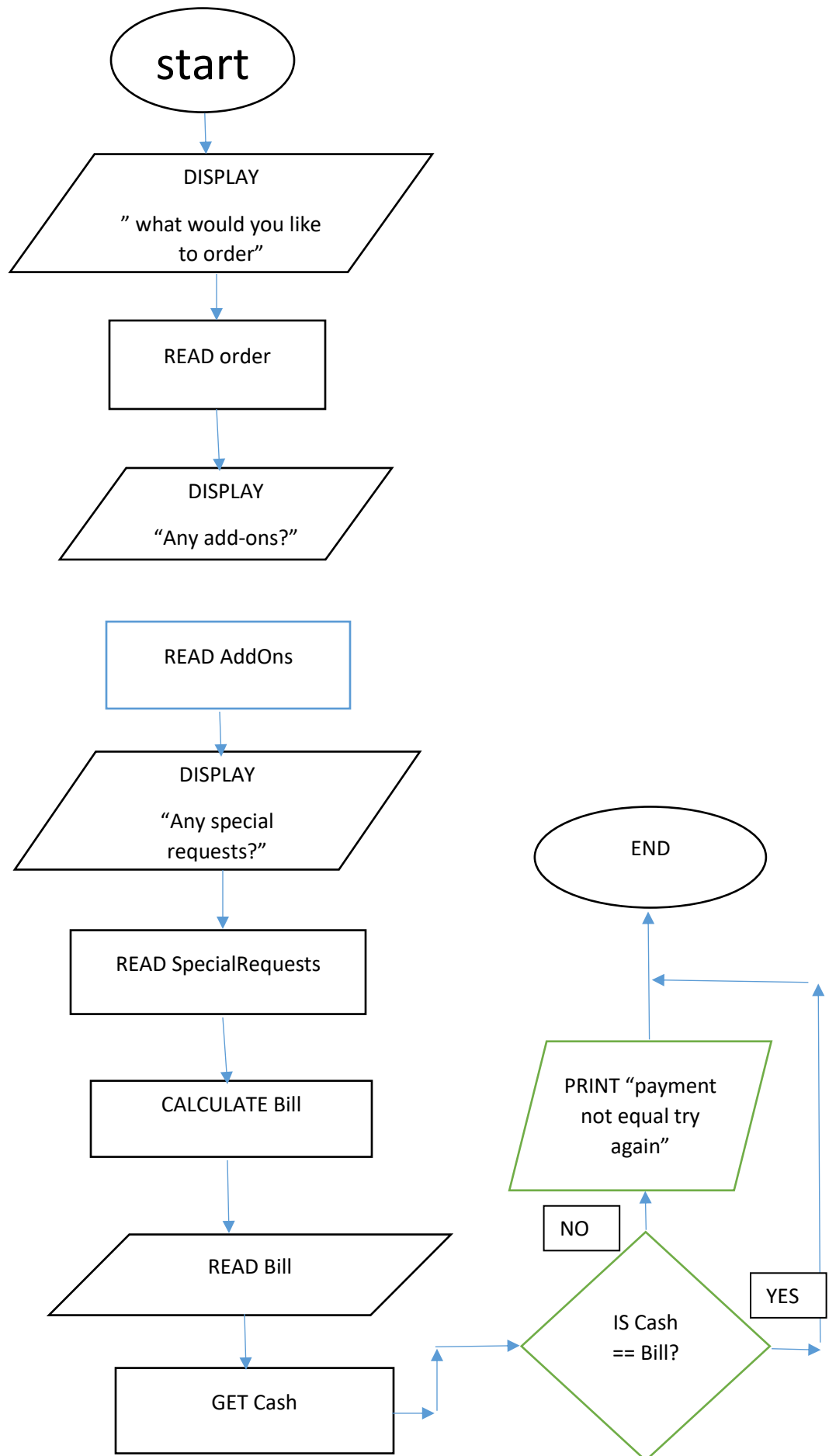1. TASK # 1
2.
3. START
4. DISPLAY "What would you like to order?"
5. READ Order
6. DISPLAY "Any add-ons?"
7. READ AddOns
8. DISPLAY "Any special requests"
9. READ SpecialRequests
10. CALCULATE Bill
11. GET Cash
12. END

1. ALGORITHM
2. Ask the user about what he wants to order
3. Read order
4. Ask the user if he wants any add-ons
5. Read add-ons
6. Ask the user for special requests
7. Read special requests
8. Calculate bill
9. GET cash

1   FLOWCHART

2

3

```
                    start

                 DISPLAY
            " what would you like
                 to order"

                READ order

                 DISPLAY
              "Any add-ons?"

               READ AddOns

                 DISPLAY
               "Any special
                requests?"

            READ SpecialRequests

              CALCULATE Bill

                READ Bill                          END

                                        PRINT "payment
                                         not equal try
                                            again"

                                    NO
                GET Cash                    IS Cash        YES
                                            == Bill?
```
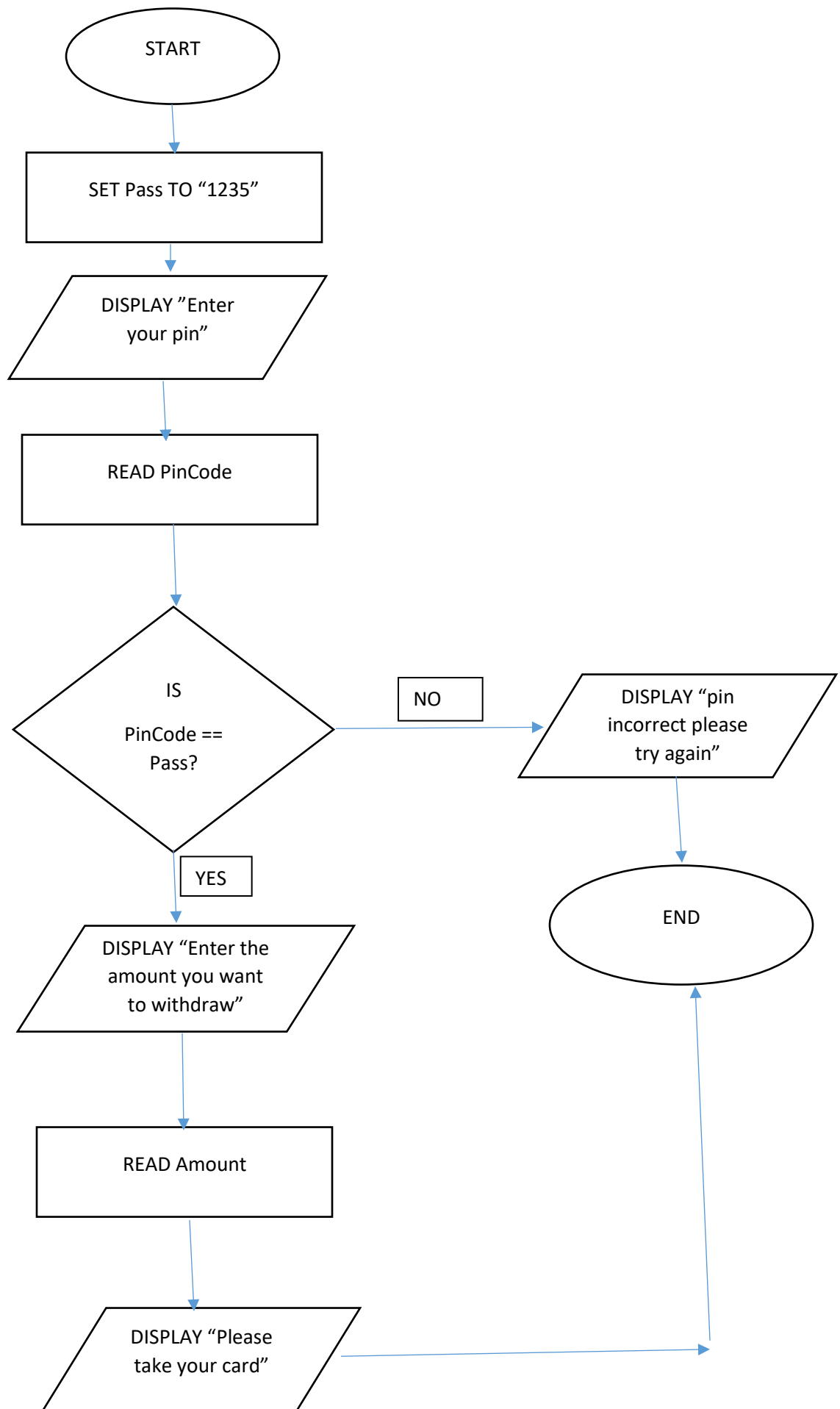
1   TASK # 2

2   START

3   SET Pass = "12345"

4   DISPLAY "Enter your password"

5   READ passkey

6   IF Pass == passkey THEN

7   DISPLAY "Enter the amount you want to withdraw"

8   READ Amount

9   DISPLAY "Take your card"

10   OUTPUT Cash

11   ELSE

12   DISPLAY "Password incorrect"

13

14   ALGORITHM

15   SET the password to "1234"

16   OUTPUT the user to enter pin

17   Read pin

18   Is password equal to pin

19   If no output "Try again"

20   If yes output "enter the amount you want to withdraw"

21   Read Amount

22   Output Amount

1

2

START

SET Pass TO "1235"

DISPLAY "Enter your pin"

READ PinCode

IS
PinCode == Pass?

NO

DISPLAY "pin incorrect please try again"

YES

DISPLAY "Enter the amount you want to withdraw"
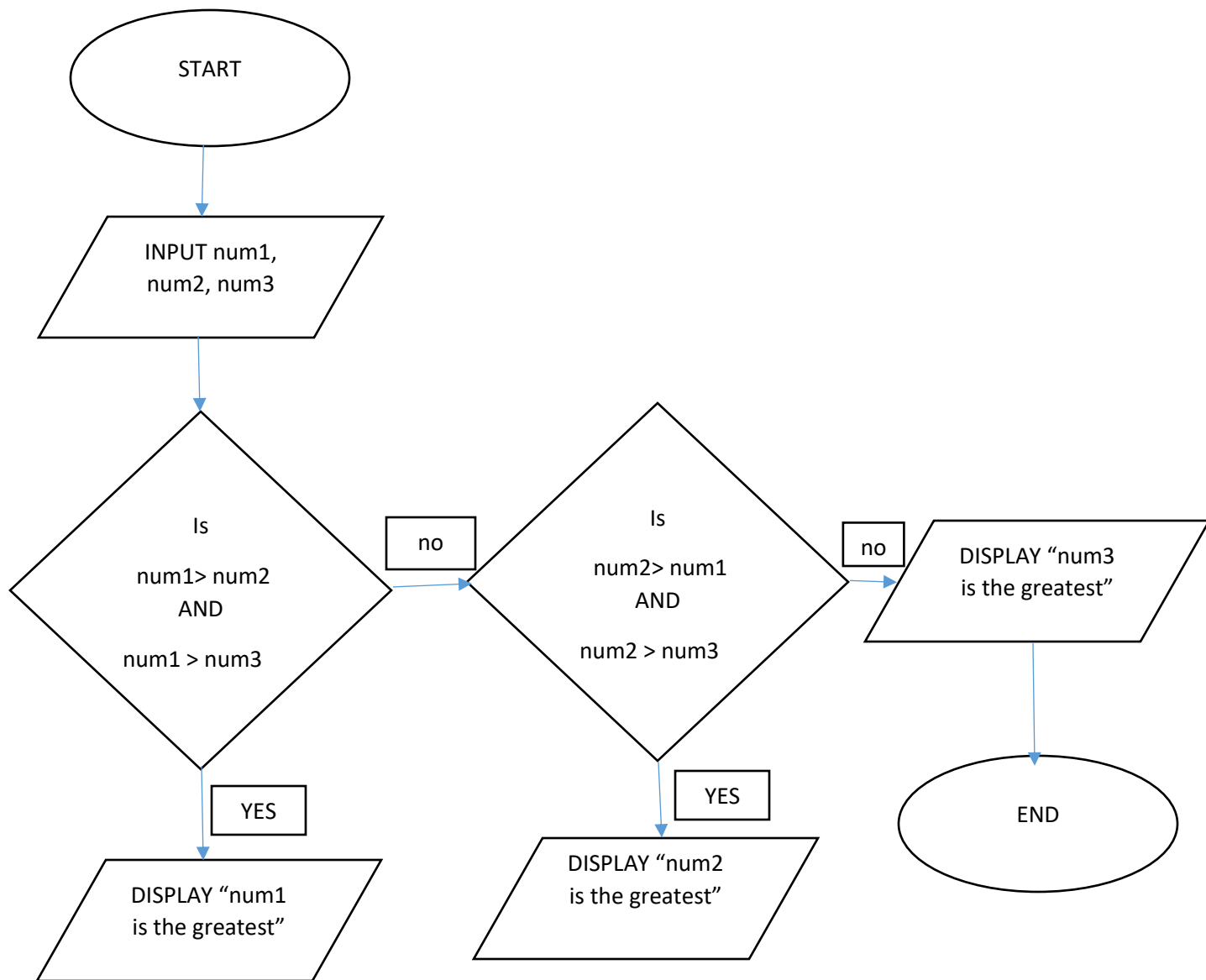
READ Amount

DISPLAY "Please take your card"

END

1 TASK # 3 (PSEUDOCODE)

2 READ num1, num2, num3

3 IF num1 > num 2 AND num1 > num3 Then

4 Print "num1 is the greatest"

5 Else if num2 > num1 and num2 > num3

6   Print"num2 is the greatest"

7 Else print "num3 is the greatest"

8

9 ALGORITHM

10 Ask the user for 3 numbers

11 Compare the first number with then last two numbers

12 If it is the greatest print num1 is the greatest

13 Else compare num2 with other 2 numbers

14 If it is the greatest print num2 is the greatest

15 Else display num3 is the greatest
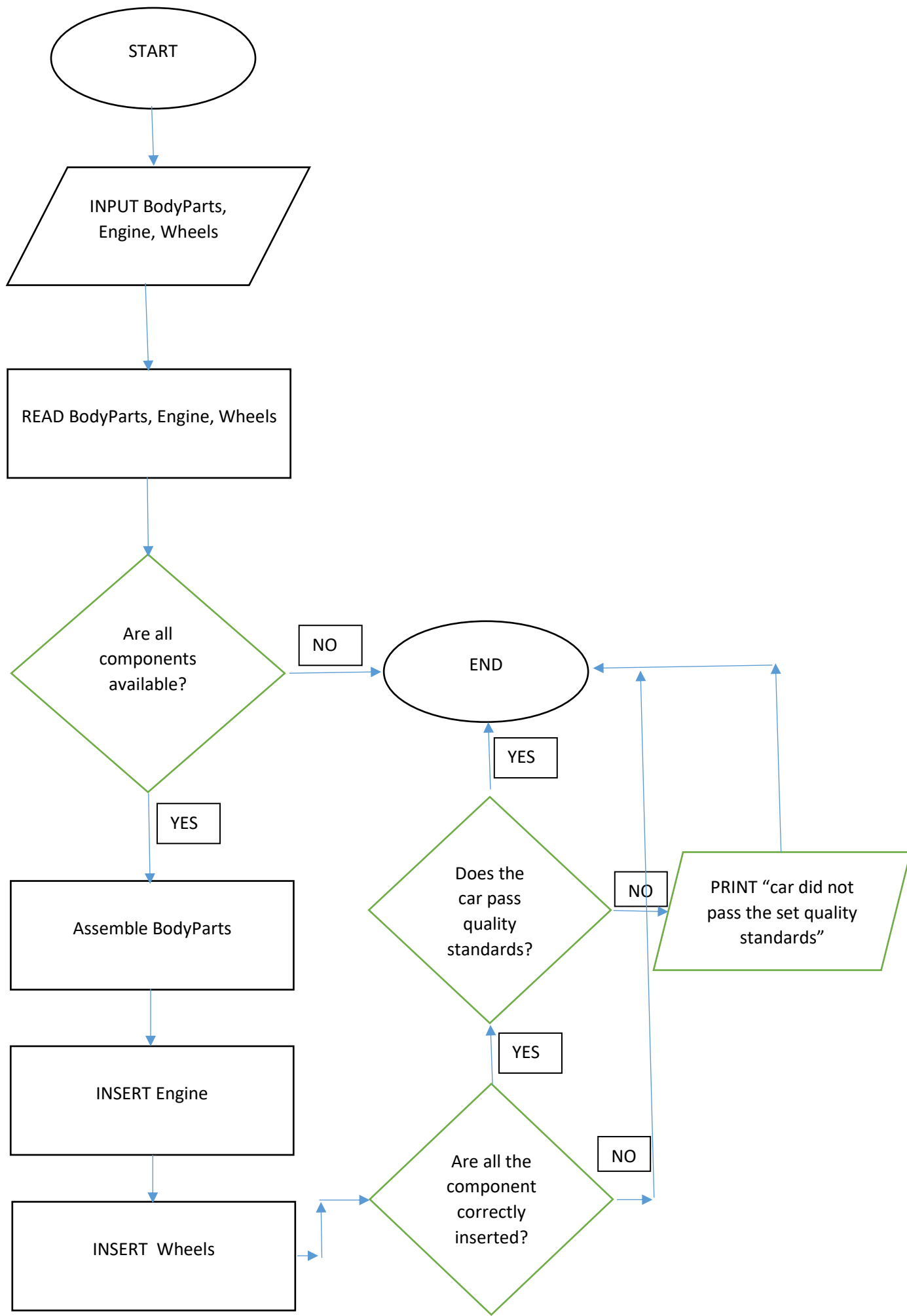
16

1

START

INPUT num1, num2, num3

Is
num1> num2
AND
num1 > num3

no

Is
num2> num1
AND
num2 > num3

no

DISPLAY "num3 is the greatest"

YES

DISPLAY "num1 is the greatest"

YES

DISPLAY "num2 is the greatest"

END

TASK # 4

1.febuary

2.march

4.april

5.may

6.june

7.july

8.august

9.september

10.october

11.november

12.decemember

Ask the user to input a number

If number is greater than 12 and less than 1 REPEAT from step 1

Compare with numbers and return month

1. TASK # 5

2. READ num1, num2, operator

3. If operator == '+' then

4. Result = num1 + num2

5. else if operator = '-'

6. Result = num1 – num2

7. Else DISPLAY "invalid operator"

8. DISPLAY Result

1	START
2	
3	INPUT BodyParts, Engine, Wheels
4	
5	
6	
7	READ BodyParts, Engine, Wheels
8	
9	
10	
11	
12	Are all components available? — NO → END
13	
14	YES
15	
16	
17	Assemble BodyParts
18	Does the car pass quality standards? — NO → PRINT "car did not pass the set quality standards"
19	
20	INSERT Engine
21	
22	Are all the component correctly inserted? — NO

START

INPUT BodyParts, Engine, Wheels

READ BodyParts, Engine, Wheels

Are all components available?

NO

END

YES

Assemble BodyParts

Does the car pass quality standards?

YES

NO

PRINT "car did not pass the set quality standards"

INSERT Engine

YES

INSERT Wheels

Are all the component correctly inserted?

NO

1   START

2   Ask the user to input 2 numbers

3   Ask the user to input an operator

4   If the operator is "+"

5   Add the two numbers

6   If the operator is "-"

7   Subtract the two numbers

8   END

9

10   TASK # 9

11   We use .gitignore to:

12   1. Reduce size of the repository.

13   2. To protect files containing sensitive information.

14   3. keeps your project cleaner and more focused.

15

16

17

18

19

20

21

22

1 TASK #10

2 ALGORITHM: An algorithm is a set of obvious, logical, and sequential
3 steps that solve a specific problem. A good algorithm is clear and
4 ambiguous, it is finite, and it is efficient.

5

6 PSEUDOCODE: Pseudocode uses program logic in human readable
7 form. It uses common structures such as loops, conditionals, and
8 function calls to outline the logic clearly. There is no definite. There is
9 no strict rules like in programming languages.