Algorithms:

Exercises

Apolo Reynoso Ruiz

Project Overview:

For each exercise described below, apply the following four points:

• Prepare the analysis of the exercise

• Elaborate the process or design of the solution

• Prepare the desktop proof of the process or design of the solution.

• Prepare the exercise flowchart

Standards:

- Analysis
- Pseudocode
- Flowcharts
- Desktop proof

Task: Calculate the area of a rectangle.

Analysis exercise 1:

To calculate the area of a rectangle, we need the length and width of the rectangle. The formula to find the area of a rectangle is length multiplied by width.

Process / design:

1. Obtain the length and width of the rectangle from the user.
2. Multiply the length by the width to find the area.
3. Display the calculated area to the user.

Pseudocode:

```
1. Prompt the user to enter the length of the rectangle
2. Store the value in a variable called length.
3. Prompt the user to enter the width of the rectangle
4. Store the value in a variable called width.
5. Calculate the area by multiplying length by width and store the result in a variable
   called area.
6. Display the area to the user.
```

Desktop proof of solution:

Let’s say the length by the user is 5 units and the width is 3 units:

length = 5

width = 3

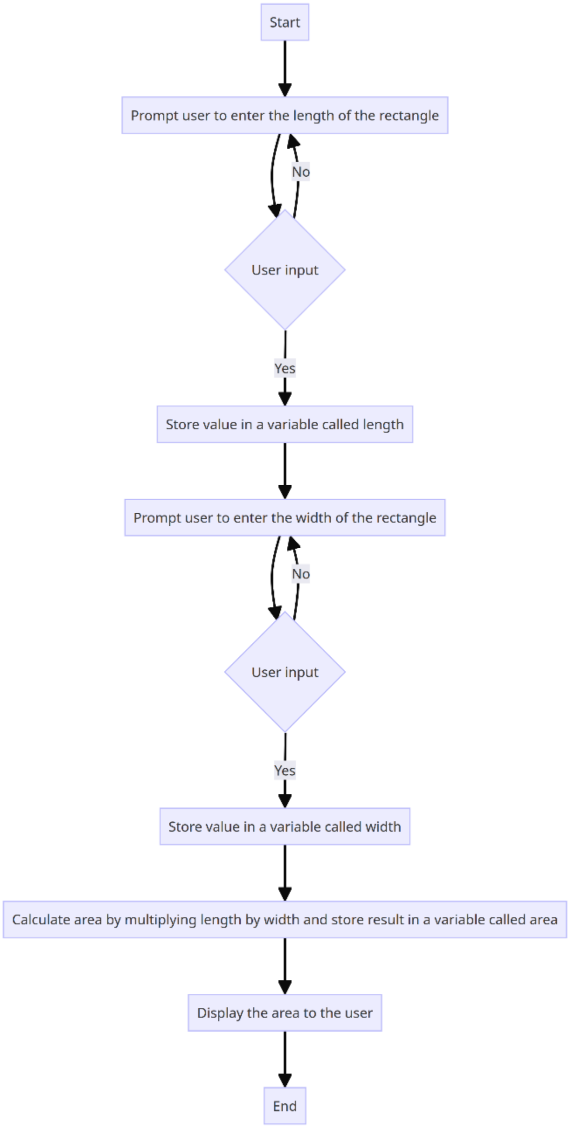
area = length * width

= 5 * 3

= 15 square units

The area of the rectangle is 15 square units.

Flowchart:


Start
Prompt user to enter the length of the rectangle
No
User input
Yes
Store value in a variable called length
Prompt user to enter the width of the rectangle
No
User input
Yes
Store value in a variable called width
Calculate area by multiplying length by width and store result in a variable called area
Display the area to the user
End

Task: Calculate the perimeter of a circle.

Analysis exercise 2:

To calculate the perimeter of a circle, we need the radius of the circle. The perimeter of a circle is also known as its circumference, which is found using the formula:

2×π×radius.

Process / design:

1. Obtain the radius of the circle from the user.
2. Calculate the perimeter (circumference) using the formula.
3. Display the calculated perimeter to the user.

Pseudocode:

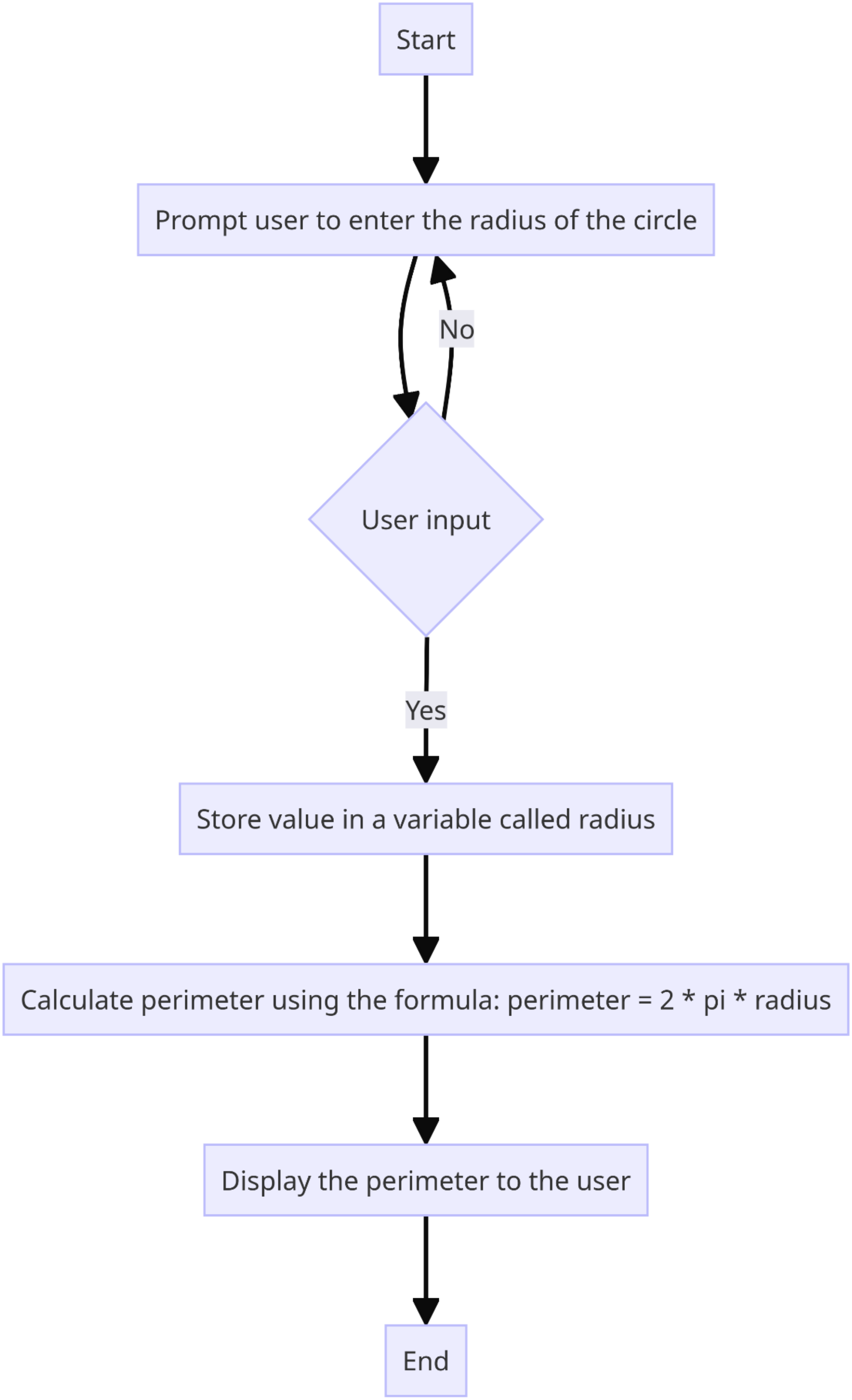
```
1. Prompt the user to enter the radius of the circle and store it in a variable called radius.
2. Calculate the perimeter using the formula: perimeter = 2 * pi * radius.
3. Display the perimeter to the user.
```

Desktop proof of solution:

Let's say the radius provided by the user is 4 units.

radius = 4
perimeter = 2 * pi * radius
= 2 * 3.14159 * 4
≈ 25.13274 units

Flowchart:


Start
Prompt user to enter the radius of the circle
No
User input
Yes
Store value in a variable called radius
Calculate perimeter using the formula: perimeter = 2 * pi * radius
Display the perimeter to the user
End

Task: Ask for two numbers and show which is greater.

Analysis exercise 3:

To determine which of two numbers is greater, we need to compare them. We'll ask the user for two numbers and then compare them to find out which one is greater.

Process / design:

1. Prompt the user to enter the first number and store it.
2. Prompt the user to enter the second number and store it.
3. Compare the two numbers to find out which one is greater.
4. Display the result to the user.

Pseudocode:

```
1. Prompt the user to enter the first number and store it in a variable called num1.
2. Prompt the user to enter the second number and store it in a variable called num2.
3. If num1 is greater than num2, display "num1 is greater".
4. Else if num2 is greater than num1, display "num2 is greater".
5. Else (if both numbers are equal), display "Both numbers are equal".
```

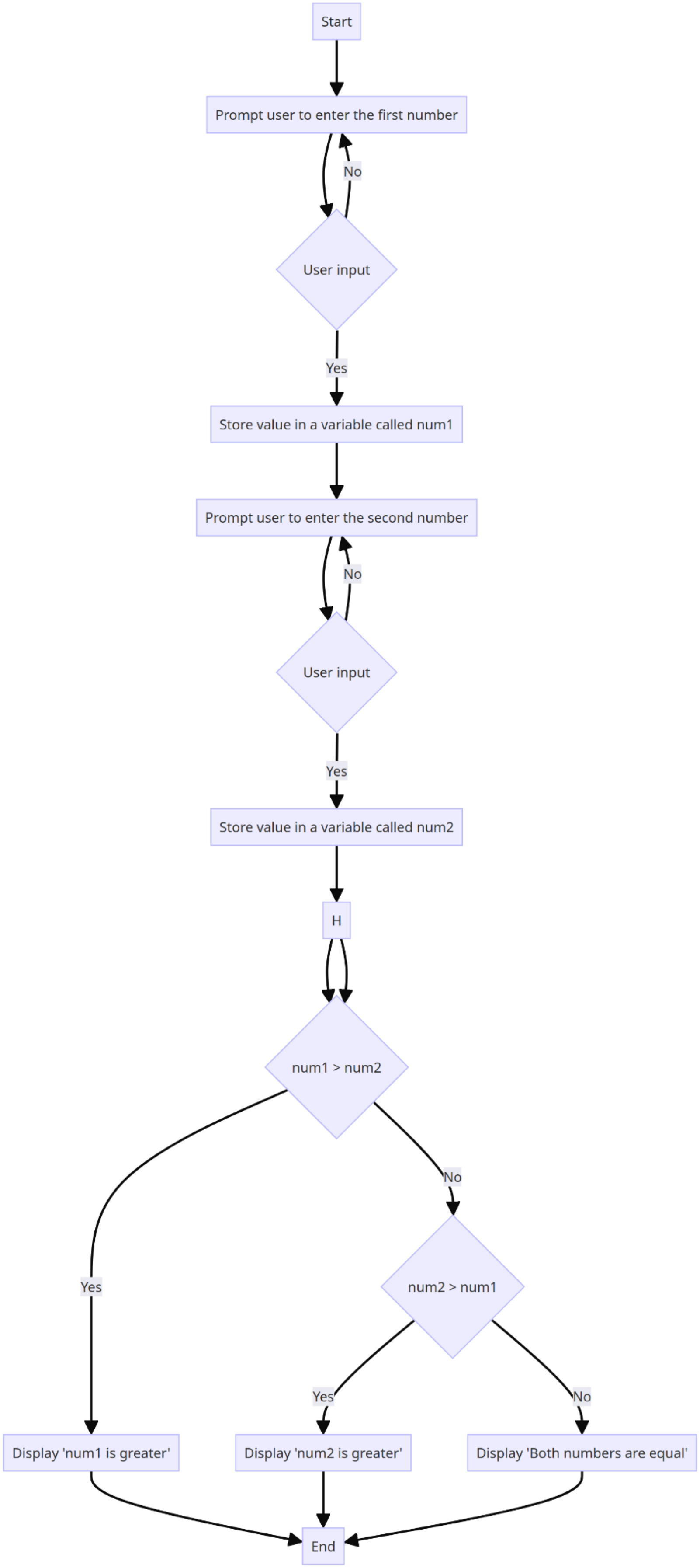
Desktop proof of solution:

Let's say the user inputs 5 as num1 and 8 as num2.

num1 = 5
num2 = 8

Since num2 is greater than num1, the output would be:
"num2 is greater"

Flowchart:


Start
Prompt user to enter the first number
No
User input
Yes
Store value in a variable called num1
Prompt user to enter the second number
No
User input
Yes
Store value in a variable called num2
H
num1 > num2
Yes
No
num2 > num1
Yes
No
Display 'num1 is greater'
Display 'num2 is greater'
Display 'Both numbers are equal'
End

Task: Ask for three numbers and show which one is smaller.

Analysis exercise 4:

To determine which of three numbers is the smallest, we need to compare them. We'll ask the user for three numbers and then compare them to find out which one is the smallest.

Process / design:

1. Prompt the user to enter the first number and store it.
2. Prompt the user to enter the second number and store it.
3. Prompt the user to enter the third number and store it.
4. Compare the three numbers to find out which one is the smallest.
5. Display the result to the user.

Pseudocode:

```
1. Prompt the user to enter the first number and store it in a variable called num1.
2. Prompt the user to enter the second number and store it in a variable called num2.
3. Prompt the user to enter the third number and store it in a variable called num3.
4. If num1 is smaller than both num2 and num3, display "num1 is the smallest".
5. Else if num2 is smaller than both num1 and num3, display "num2 is the smallest".
6. Else if num3 is smaller than both num1 and num2, display "num3 is the smallest".
7. Else (if there are two or more equal smallest numbers), display "There are multiple
smallest numbers".
```

Desktop proof of solution:

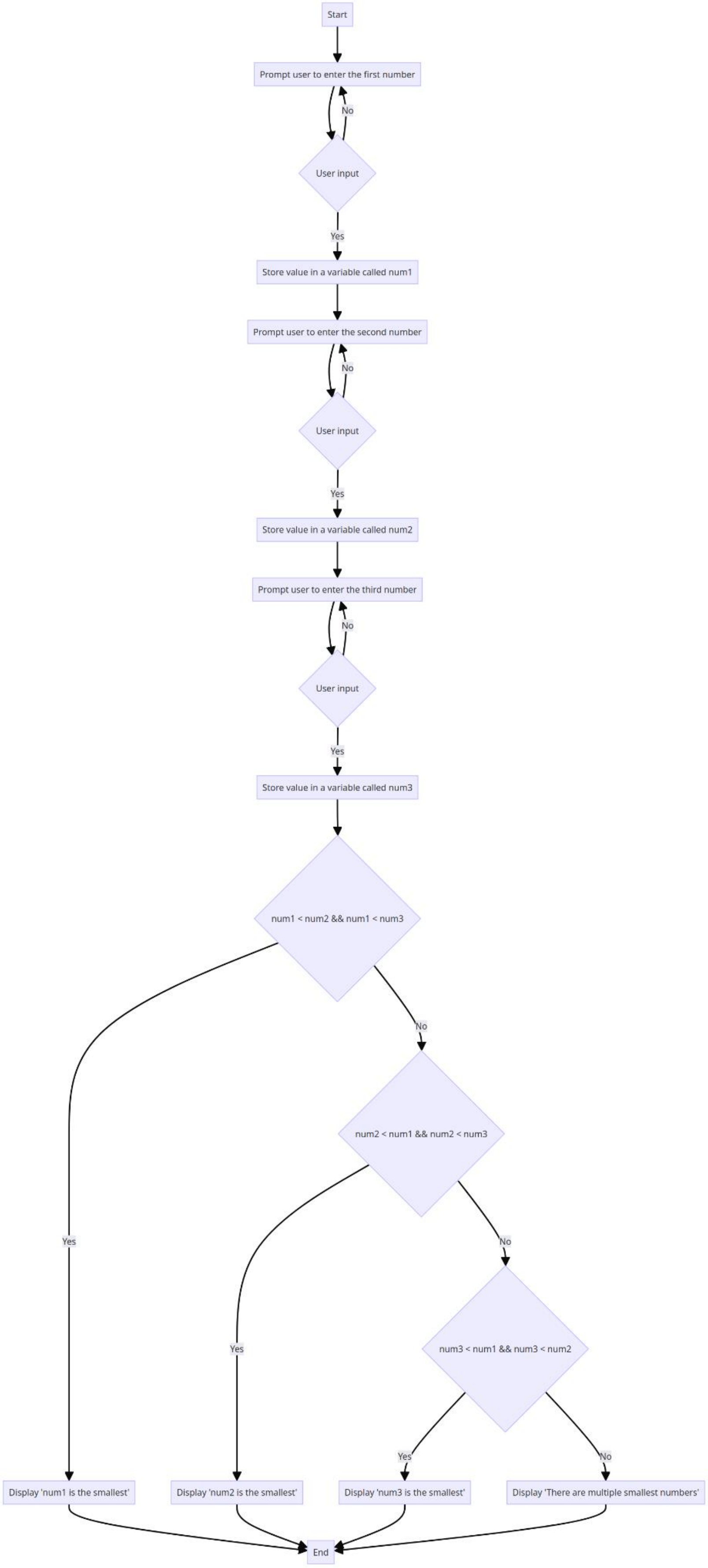
Let's say the user inputs 5, 3, and 7 as num1, num2, and num3 respectively.

num1 = 5
num2 = 3
num3 = 7

Since num2 is smaller than both num1 and num3, the output would be:

"num2 is the smallest"

Flowchart:


Start
Prompt user to enter the first number
No
User input
Yes
Store value in a variable called num1
Prompt user to enter the second number
No
User input
Yes
Store value in a variable called num2
Prompt user to enter the third number
No
User input
Yes
Store value in a variable called num3
num1 < num2 && num1 < num3
No
Yes
num2 < num1 && num2 < num3
No
Yes
num3 < num1 && num3 < num2
Yes
No
Display 'num1 is the smallest'
Display 'num2 is the smallest'
Display 'num3 is the smallest'
Display 'There are multiple smallest numbers'
End

Task: Calculate the area of a rectangle.

Analysis exercise 5:

To determine if three numbers are the same, we need to compare them. We'll ask the user for three numbers and then check if they are all the same or if there are differences among them.

Process / design:

1. Prompt the user to enter the first number and store it.
2. Prompt the user to enter the second number and store it.
3. Prompt the user to enter the third number and store it.
4. Compare the three numbers to determine if they are all the same.
5. Display the result to the user.

Pseudocode:

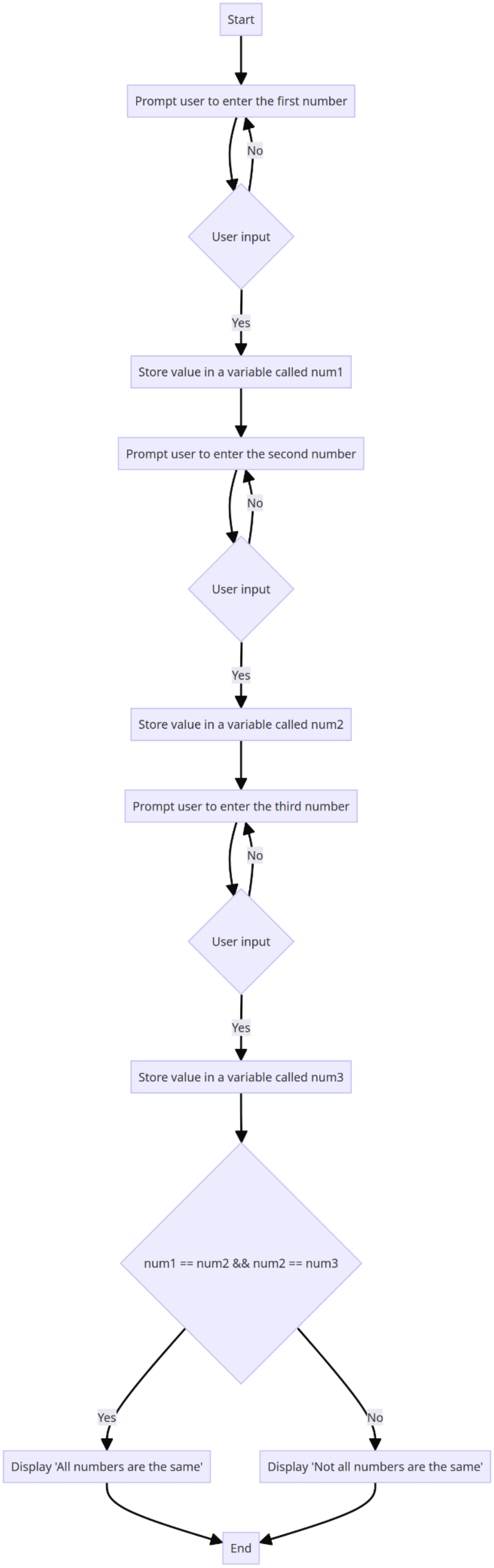
```
1. Prompt the user to enter the first number and store it in a variable called num1.
2. Prompt the user to enter the second number and store it in a variable called num2.
3. Prompt the user to enter the third number and store it in a variable called num3.
4. If num1 equals num2 and num2 equals num3, display "All numbers are the same".
5. Else, display "Not all numbers are the same".
```

Desktop proof of solution:

Let's say the user inputs 5, 5, and 5 as num1, num2, and num3 respectively.

num1 = 5
num2 = 5
num3 = 5
Since num1 equals num2 and num2 equals num3, the output would be:
"All numbers are the same"

Flowchart:


Start
Prompt user to enter the first number
No
User input
Yes
Store value in a variable called num1
Prompt user to enter the second number
No
User input
Yes
Store value in a variable called num2
Prompt user to enter the third number
No
User input
Yes
Store value in a variable called num3
num1 == num2 && num2 == num3
Yes
No
Display 'All numbers are the same'
Display 'Not all numbers are the same'
End

Task: Ask for an amount in miles and convert it to millimeters.

Analysis exercise 6:

To convert miles to kilometers, we need to know the conversion factor between miles and kilometers. The conversion factor is 1 mile = 1.60934 kilometers. We'll ask the user for an amount in miles and then perform the conversion.

Process / design:

1. Prompt the user to enter the distance in miles.
2. Multiply the distance in miles by the conversion factor to obtain the distance in kilometers.
3. Display the converted distance to the user.

Pseudocode:

```
1. Prompt the user to enter the distance in miles and store it in a variable called miles.
2. Multiply the miles by the conversion factor (1.60934) and store the result in a
variable called kilometers.
3. Display the value of kilometers to the user.
```

Desktop proof of solution:

Let's say the user inputs 10 miles.

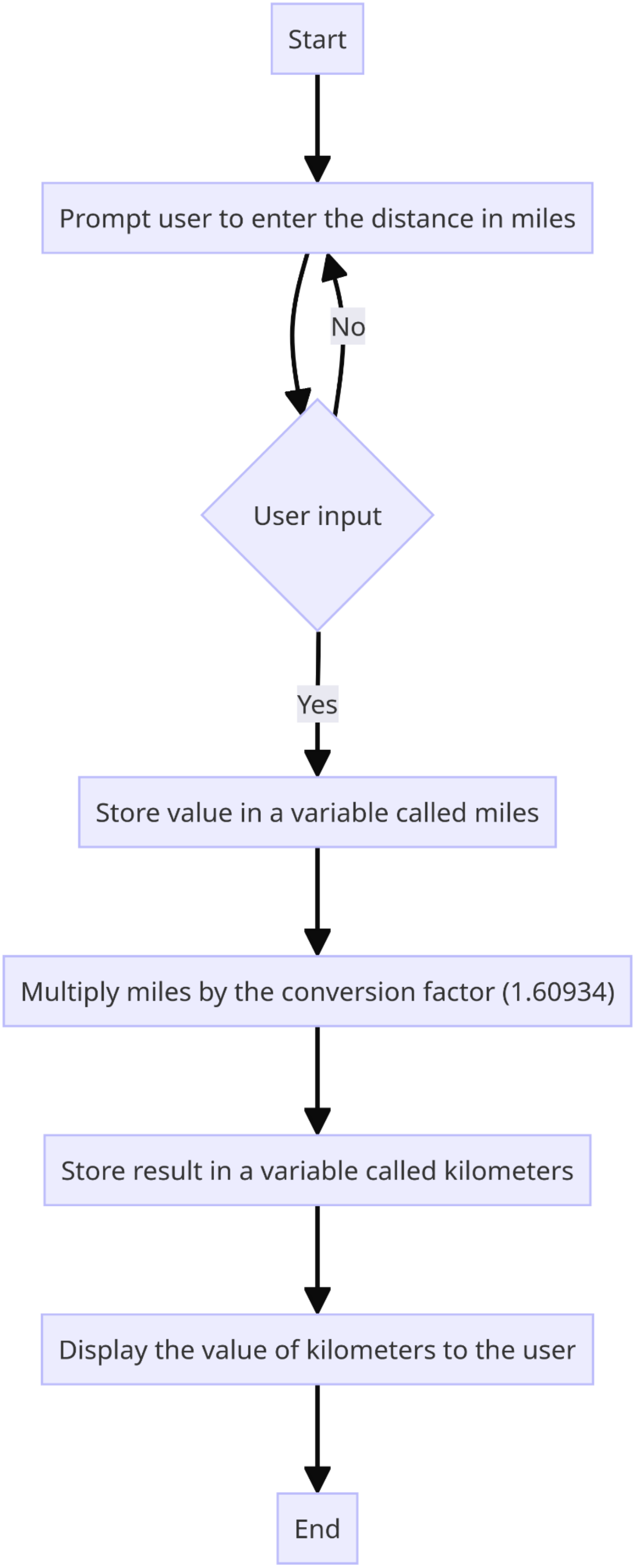
miles = 10

kilometers = miles * conversion_factor

= 10 * 1.60934
≈ 16.0934 kilometers

Thus, 10 miles is approximately equal to 16.0934 kilometers.

Flowchart:


Start
Prompt user to enter the distance in miles
No
User input
Yes
Store value in a variable called miles
Multiply miles by the conversion factor (1.60934)
Store result in a variable called kilometers
Display the value of kilometers to the user
End

Task: Ask for a quantity in degrees Celsius and convert it to Farenheit.

Analysis exercise 7:

To convert temperature from degrees Celsius to Fahrenheit, we use the formula: F=95×C+32F=59×C+32, where F represents Fahrenheit and C represents Celsius. We'll ask the user for a temperature in degrees Celsius and then perform the conversion.

Process / design:

1. Prompt the user to enter the temperature in degrees Celsius.
2. Apply the conversion formula to convert Celsius to Fahrenheit.
3. Display the converted temperature to the user.

Pseudocode:

```
1. Prompt the user to enter the temperature in degrees Celsius and store it in a variable called Celsius.
2. Calculate the temperature in Fahrenheit using the formula: Fahrenheit = (9/5) * Celsius + 32.
3. Display the value of Fahrenheit to the user.
```

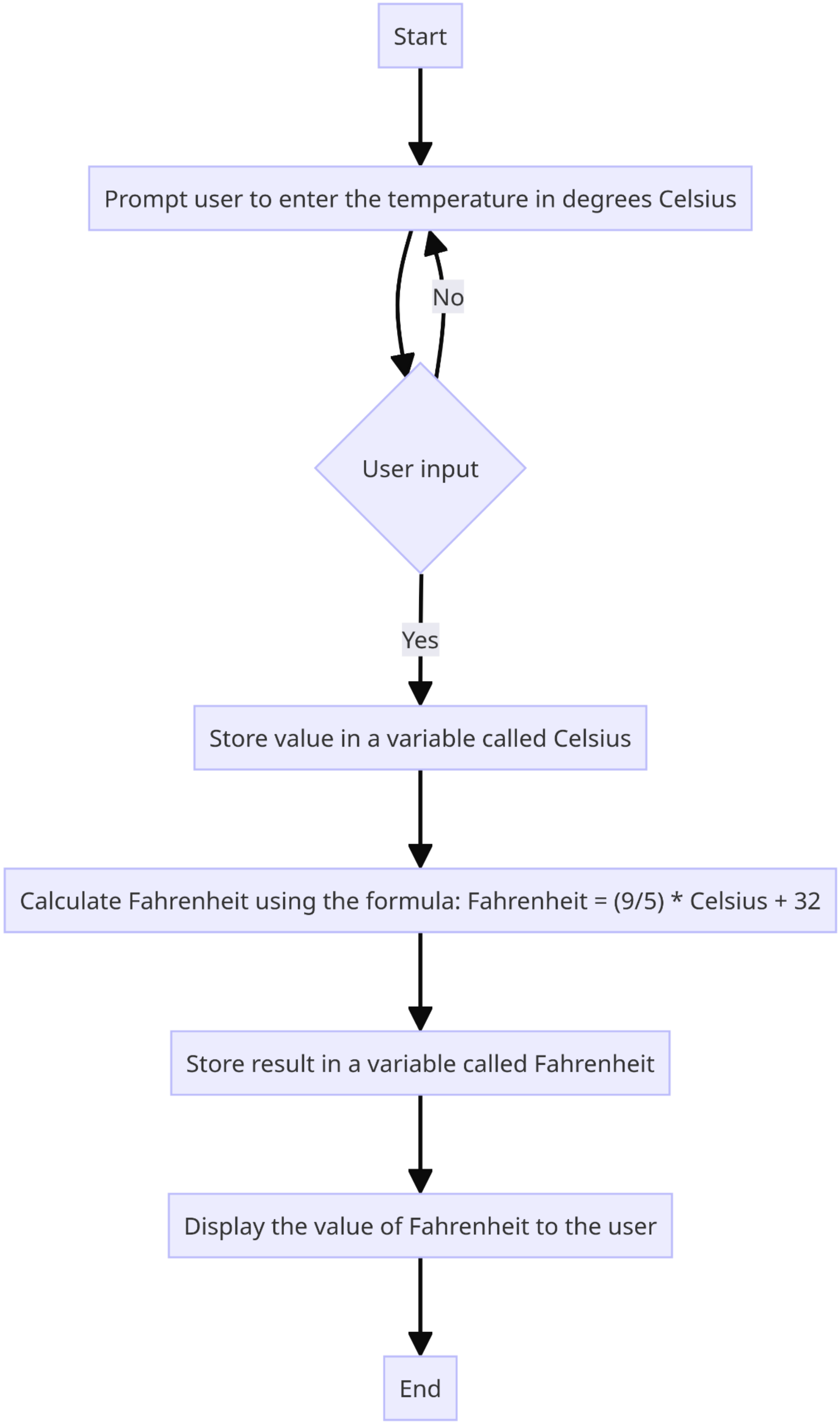
Desktop proof of solution:

Let's say the user inputs 20 degrees Celsius.

```
celsius = 20
fahrenheit = (9/5) * celsius + 32
           = (9/5) * 20 + 32
           = 36 + 32
           = 68 degrees Fahrenheit
```

Thus, 20 degrees Celsius is equal to 68 degrees Fahrenheit.

Flowchart:


Start
Prompt user to enter the temperature in degrees Celsius
No
User input
Yes
Store value in a variable called Celsius
Calculate Fahrenheit using the formula: Fahrenheit = (9/5) * Celsius + 32
Store result in a variable called Fahrenheit
Display the value of Fahrenheit to the user
End

Task: Ask for a quantity in kilometers and convert it to millimeters.

Analysis exercise 8:

To convert a quantity from kilometers to millimeters, we need to know the conversion factor between kilometers and millimeters. There are 1,000,000 millimeters in 1 kilometer. We'll ask the user for a quantity in kilometers and then perform the conversion.

Process / design:

1. Prompt the user to enter the quantity in kilometers.
2. Multiply the quantity in kilometers by the conversion factor to obtain the quantity in millimeters.
3. Display the converted quantity to the user.

Pseudocode:

```
1. Prompt the user to enter the quantity in kilometers and store it in a variable called kilometers.
2. Multiply the kilometers by the conversion factor (1,000,000) and store the result in a variable called millimeters.
3. Display the value of millimeters to the user.
```

Desktop proof of solution:

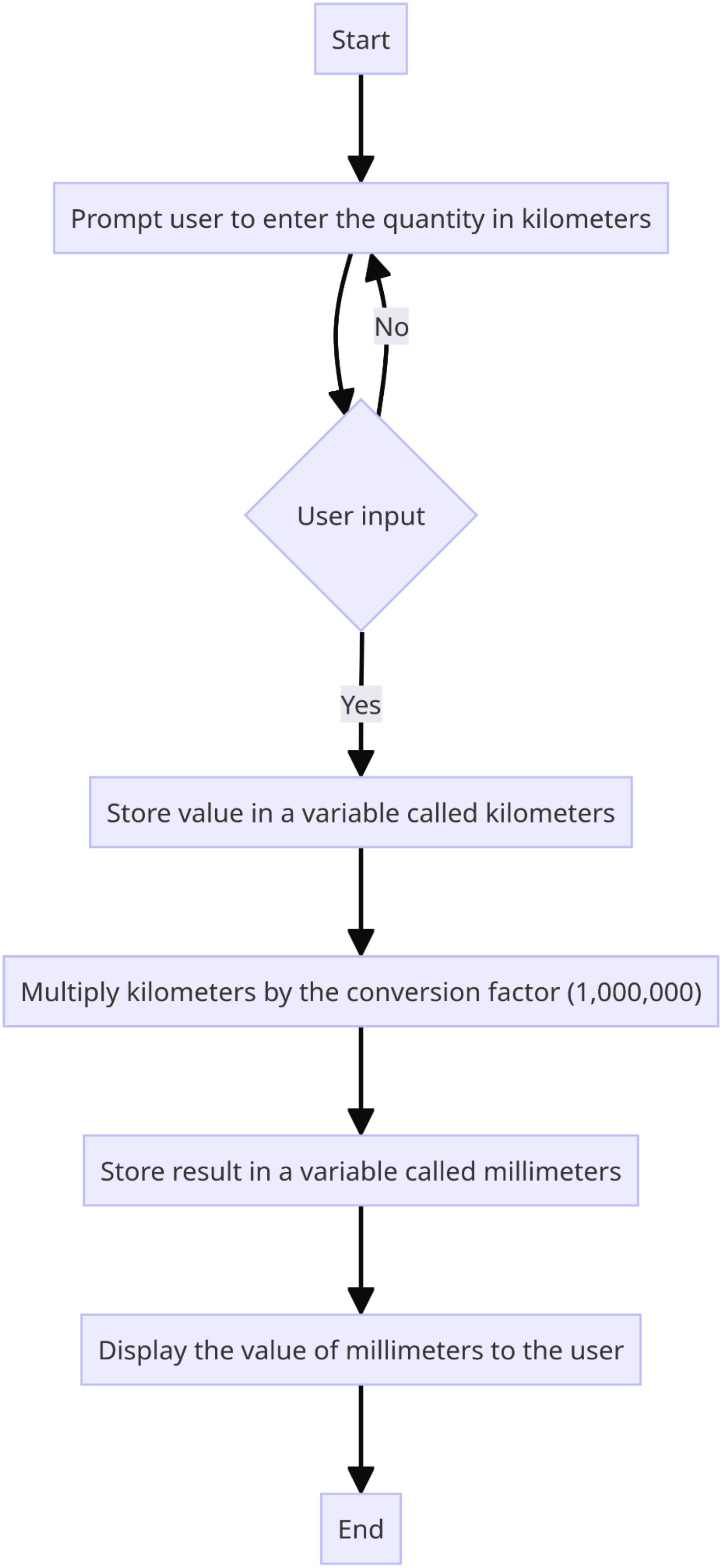
Let's say the user inputs 10 kilometers.

kilometers = 10

millimeters = kilometers * conversion_factor
= 10 * 1,000,000
= 10,000,000 millimeters

Thus, 10 kilometers is equal to 10,000,000 millimeters.

Flowchart:


Start
Prompt user to enter the quantity in kilometers
No
User input
Yes
Store value in a variable called kilometers
Multiply kilometers by the conversion factor (1,000,000)
Store result in a variable called millimeters
Display the value of millimeters to the user
End

Task: Ask for an amount in kilograms and convert it to grams.

Analysis exercise 9:

To convert an amount from kilograms to grams, we need to know the conversion factor between kilograms and grams. There are 1000 grams in 1 kilogram. We'll ask the user for an amount in kilograms and then perform the conversion.

Process / design:

1. Prompt the user to enter the amount in kilograms.
2. Multiply the amount in kilograms by the conversion factor to obtain the amount in grams.
3. Display the converted amount to the user.

Pseudocode:

```
1. Prompt the user to enter the amount in kilograms and store it in a variable called kilograms.
2. Multiply the kilograms by the conversion factor (1000) and store the result in a variable called grams.
3. Display the value of grams to the user.
```

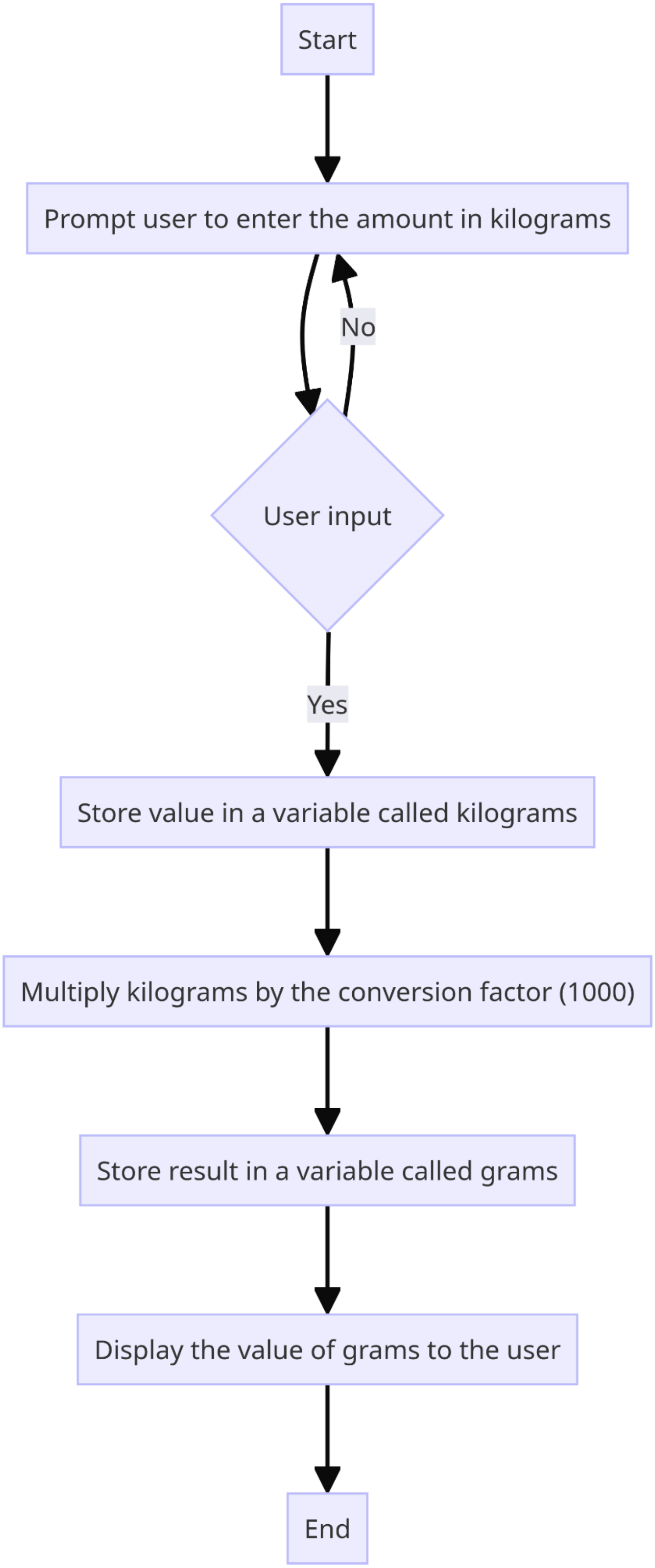
Desktop proof of solution:

Let's say the user inputs 5 kilograms.

```
kilograms = 5
grams = kilograms * conversion_factor
                  = 5 * 1000
                  = 5000 grams
```

Thus, 5 kilograms is equal to 5000 grams.

Flowchart:


Start
Prompt user to enter the amount in kilograms
No
User input
Yes
Store value in a variable called kilograms
Multiply kilograms by the conversion factor (1000)
Store result in a variable called grams
Display the value of grams to the user
End

Task: Ask for a quantity in deciliters and convert it to liters.

Analysis exercise 10:

To convert a quantity from deciliters to liters, we need to know the conversion factor between deciliters and liters. There are 0.1 liters in 1 deciliter. We'll ask the user for a quantity in deciliters and then perform the conversion.

Process / design:

1. Prompt the user to enter the quantity in deciliters.
2. Multiply the quantity in deciliters by the conversion factor to obtain the quantity in liters.
3. Display the converted quantity to the user.
4. Prompt the user to enter the quantity in deciliters.
5. Multiply the quantity in deciliters by the conversion factor to obtain the quantity in liters.
6. Display the converted quantity to the user.

Pseudocode:

1. ***Prompt the user to enter the quantity in deciliters and store it in a variable called deciliters.***
2. ***Multiply the deciliters by the conversion factor (0.1) and store the result in a variable called liters.***
3. ***Display the value of liters to the user.***

Desktop proof of solution:

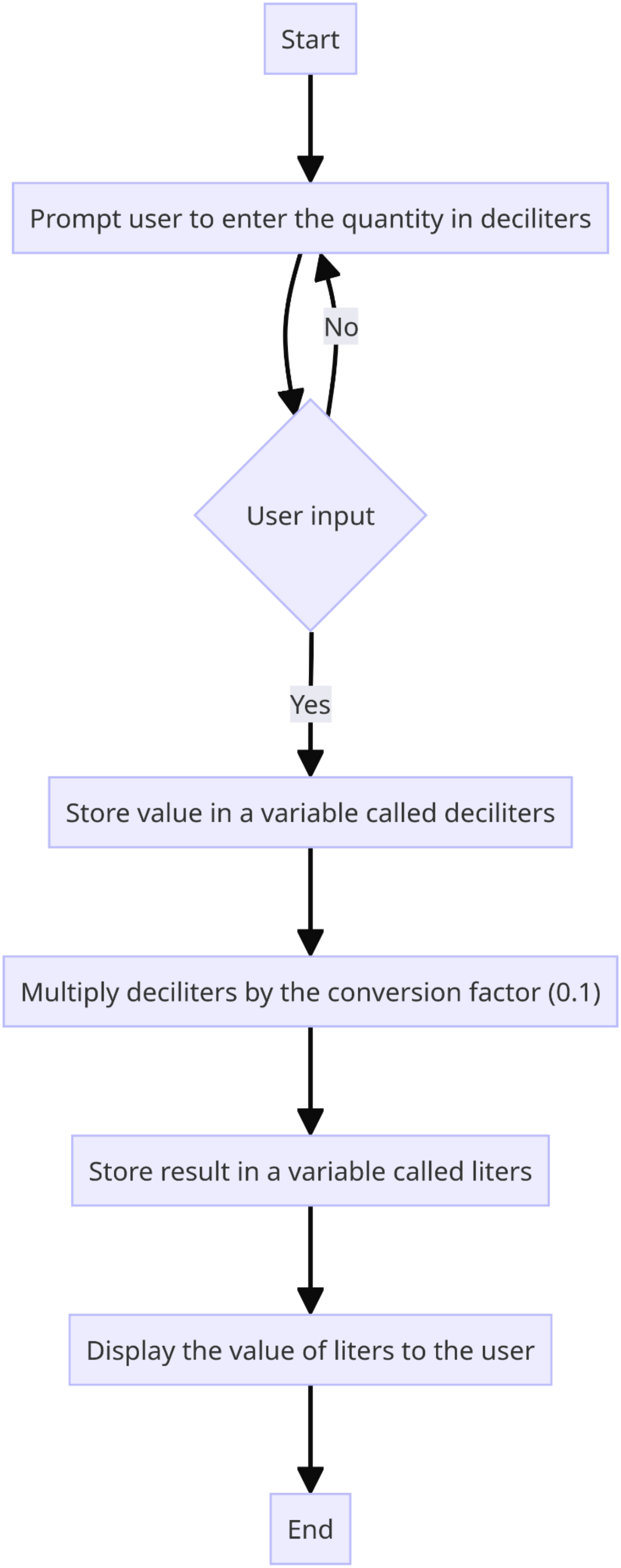
Let's say the user inputs 20 deciliters.

deciliters = 20

liters = deciliters * conversion_factor
= 20 * 0.1
= 2 liters

Thus, 20 deciliters are equal to 2 liters.

Flowchart:


Start
Prompt user to enter the quantity in deciliters
No
User input
Yes
Store value in a variable called deciliters
Multiply deciliters by the conversion factor (0.1)
Store result in a variable called liters
Display the value of liters to the user
End

Task: Ask for three numbers and show the result of the addition, subtraction, multiplication and division of the quantities entered.

Analysis exercise 11: To perform addition, subtraction, multiplication, and division of three numbers, we need to ask the user for these three numbers and then carry out the required operations.

Process / design:

1. Prompt the user to enter the first number and store it.
2. Prompt the user to enter the second number and store it.
3. Prompt the user to enter the third number and store it.
4. Perform addition, subtraction, multiplication, and division operations on the three numbers.
5. Display the results of these operations to the user.

Pseudocode:

```
1. Prompt the user to enter the first number and store it in a variable called num1.
2. Prompt the user to enter the second number and store it in a variable called num2.
3. Prompt the user to enter the third number and store it in a variable called num3.
4. Perform addition: result_addition = num1 + num2 + num3.
5. Perform subtraction: result_subtraction = num1 - num2 - num3.
6. Perform multiplication: result_multiplication = num1 * num2 * num3.
7. Perform division: result_division = num1 / num2 / num3.
8. Display the results of addition, subtraction, multiplication, and division to the user.
```

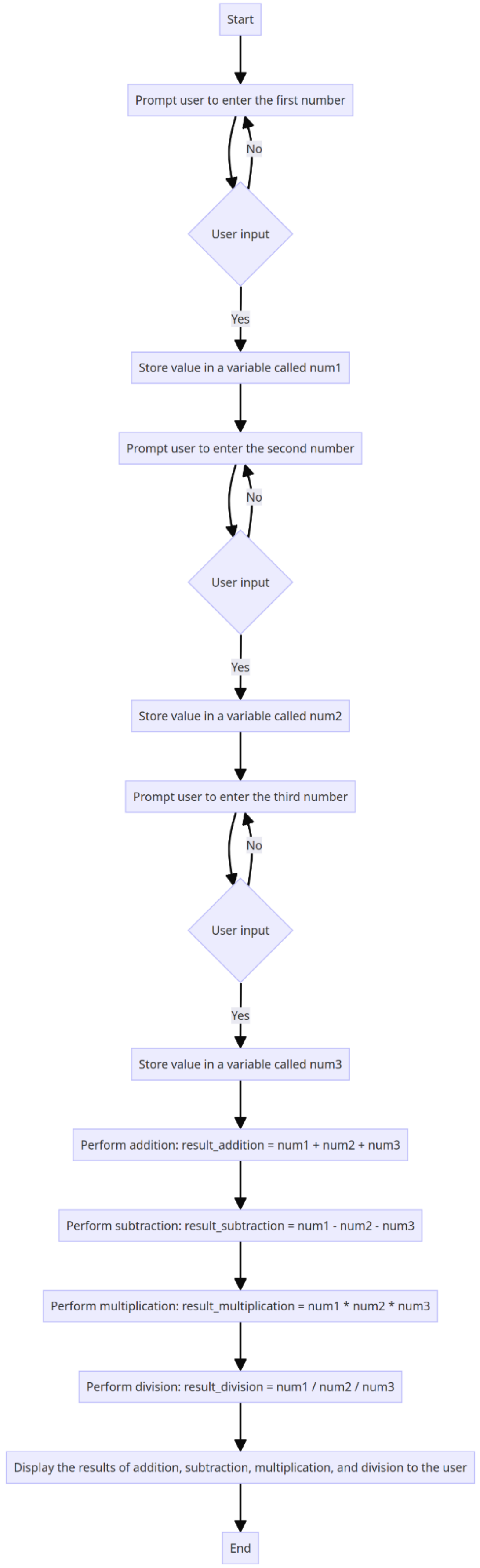
Desktop proof of solution:

Let's say the user inputs 5, 3, and 2 as num1, num2, and num3 respectively.

num1 = 5
num2 = 3
num3 = 2

result_addition = 5 + 3 + 2 = 10
result_subtraction = 5 - 3 - 2 = 0
result_multiplication = 5 * 3 * 2 = 30
result_division = 5 / 3 / 2 ≈ 0.83

Flowchart:


Start
Prompt user to enter the first number
No
User input
Yes
Store value in a variable called num1
Prompt user to enter the second number
No
User input
Yes
Store value in a variable called num2
Prompt user to enter the third number
No
User input
Yes
Store value in a variable called num3
Perform addition: result_addition = num1 + num2 + num3
Perform subtraction: result_subtraction = num1 - num2 - num3
Perform multiplication: result_multiplication = num1 * num2 * num3
Perform division: result_division = num1 / num2 / num3
Display the results of addition, subtraction, multiplication, and division to the user
End

Task: Ask for a number and show if it is even or not.

Analysis exercise 12: To determine if a number is even or odd, we need to check if it is divisible by 2 without any remainder. If the number is divisible by 2, it is even; otherwise, it is odd

Process / design:

1. Prompt the user to enter a number.
2. Check if the number is divisible by 2.
3. If the number is divisible by 2, it's even; otherwise, it's odd.
4. Display the result to the user.

Pseudocode:

```
1. Prompt the user to enter a number and store it in a variable called num.
2. If num modulo 2 equals 0:
     - Display "The number is even".
   Else:
     - Display "The number is odd".
```

Desktop proof of solution:

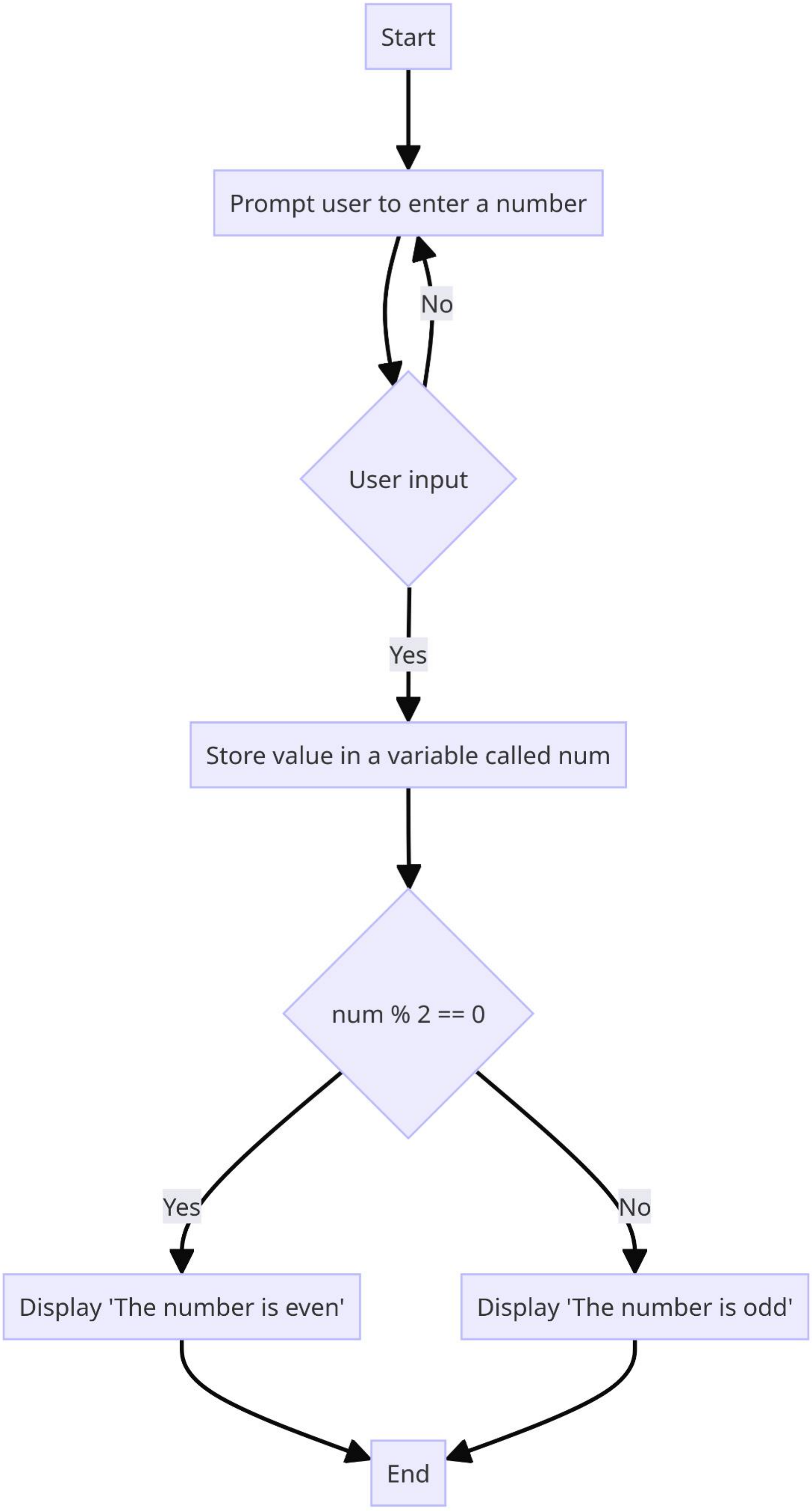
Let's say the user inputs 6.

num = 6

6 % 2 = 0

Since 6 is divisible by 2 without any remainder, it's even.

Flowchart:


Start
Prompt user to enter a number
No
User input
Yes
Store value in a variable called num
num % 2 == 0
Yes
No
Display 'The number is even'
Display 'The number is odd'
End

Task: Ask for a number from 1 to 7 and say the corresponding day of the week starting with Monday as the number 1 and ending with Sunday as the number 7.

Analysis exercise 13: This exercise requires taking a number input from the user within the range of 1 to 7 and then associating each number with a day of the week, starting with Monday as 1 and ending with Sunday as 7.

Process / design:

1. Prompt the user to enter a number between 1 and 7.
2. Based on the input number, associate it with a day of the week.
3. Display the corresponding day to the user.

Pseudocode:

```
1. Prompt the user to enter a number between 1 and 7 and store it in a variable called num.
2. If num equals 1, display "Monday".
3. Else if num equals 2, display "Tuesday".
4. Else if num equals 3, display "Wednesday".
5. Else if num equals 4, display "Thursday".
6. Else if num equals 5, display "Friday".
7. Else if num equals 6, display "Saturday".
8. Else if num equals 7, display "Sunday".
9. Else, display an error message indicating the input is out of range.
```

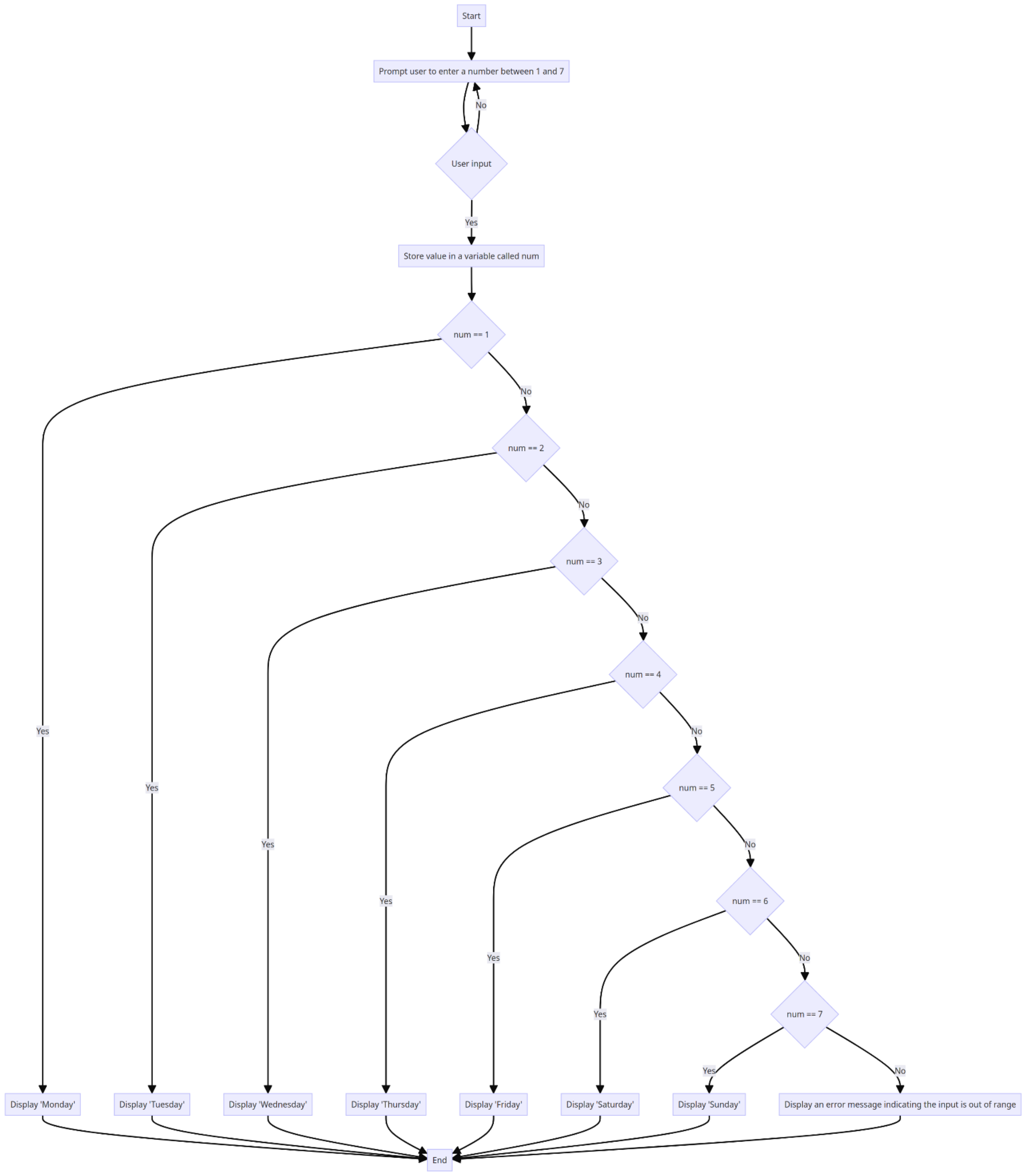
Desktop proof of solution:

Let's say the user inputs 3

num = 3

Since num equals 3, the output would be:
"Wednesday"

Flowchart:


Start
Prompt user to enter a number between 1 and 7
No
User input
Yes
Store value in a variable called num
num == 1
No
num == 2
No
num == 3
No
num == 4
No
num == 5
No
num == 6
No
num == 7
Yes
Yes
Yes
Yes
Yes
Yes
Yes
No
Display 'Monday'
Display 'Tuesday'
Display 'Wednesday'
Display 'Thursday'
Display 'Friday'
Display 'Saturday'
Display 'Sunday'
Display an error message indicating the input is out of range
End

Task: Ask for 3 numbers and show them from smallest to largest.

Analysis exercise 14: This exercise involves taking three numbers as input from the user and then displaying them in ascending order, from the smallest to the largest.

Process / design:

1. Prompt the user to enter three numbers.
2. Compare the three numbers to find the smallest, middle, and largest numbers.
3. Arrange the numbers in ascending order.
4. Display the numbers in ascending order to the user.

Pseudocode:

```
1. Prompt the user to enter three numbers and store them in variables num1, num2, and num3.
2. Find the smallest number among num1, num2, and num3 and store it in a variable called smallest.
3. Find the largest number among num1, num2, and num3 and store it in a variable called largest.
4. Calculate the middle number as the sum of num1, num2, and num3 minus the sum of smallest and largest.
5. Display the numbers in ascending order: smallest, middle, and largest.
```

Desktop proof of solution:

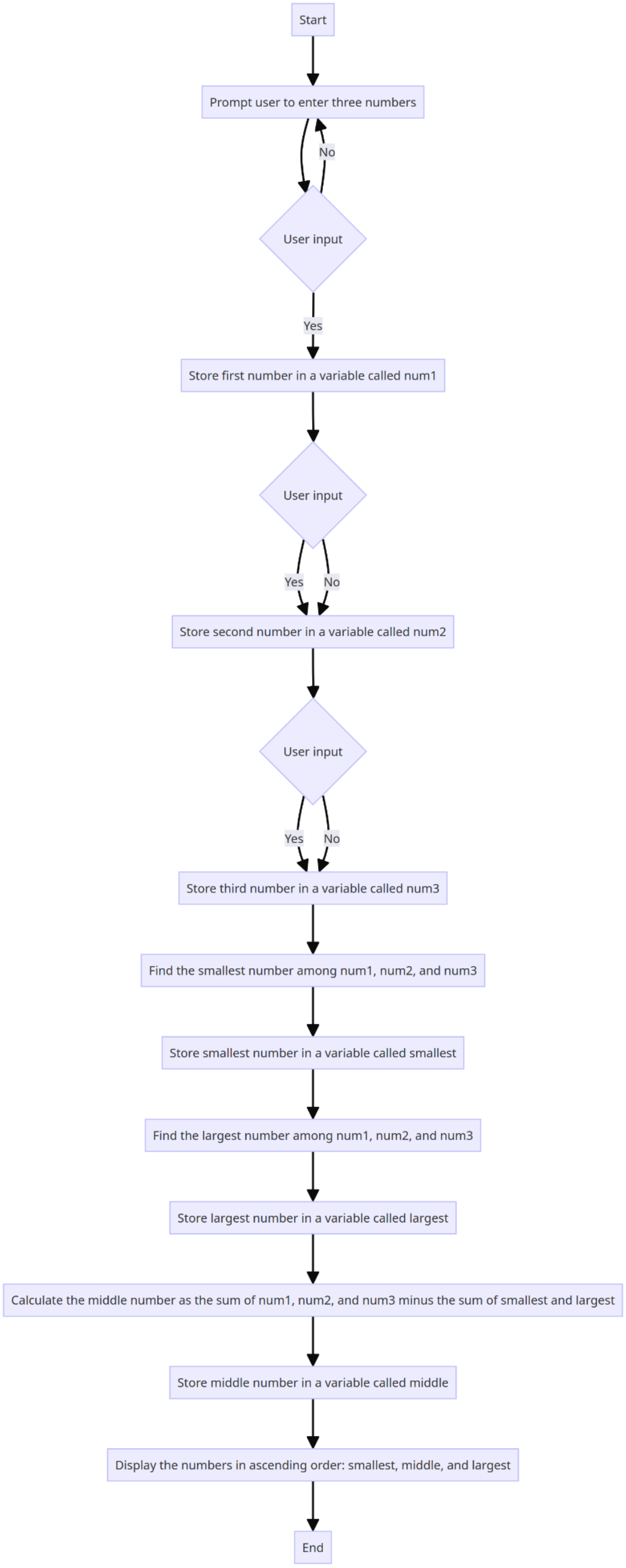
Let's say the user inputs 5, 3, and 8.

num1 = 5
num2 = 3
num3 = 8

smallest = 3
middle = 5
largest = 8

The numbers in ascending order are: 3, 5, 8

Flowchart:


Start
Prompt user to enter three numbers
No
User input
Yes
Store first number in a variable called num1
User input
Yes
No
Store second number in a variable called num2
User input
Yes
No
Store third number in a variable called num3
Find the smallest number among num1, num2, and num3
Store smallest number in a variable called smallest
Find the largest number among num1, num2, and num3
Store largest number in a variable called largest
Calculate the middle number as the sum of num1, num2, and num3 minus the sum of smallest and largest
Store middle number in a variable called middle
Display the numbers in ascending order: smallest, middle, and largest
End

Task: Ask for 3 numbers and show them from greatest to least.

Analysis exercise 15: This exercise involves taking three numbers as input from the user and then displaying them in descending order, from greatest to least.

Process / design:

1. Prompt the user to enter three numbers.
2. Compare the three numbers to find the greatest, middle, and least numbers.
3. Arrange the numbers in descending order.
4. Display the numbers in descending order to the user.

Pseudocode:

```
1. Prompt the user to enter three numbers and store them in variables num1, num2, and num3.
2. Find the greatest number among num1, num2, and num3 and store it in a variable called greatest.
3. Find the least number among num1, num2, and num3 and store it in a variable called least.
4. Calculate the middle number as the sum of num1, num2, and num3 minus the sum of greatest and least.
5. Display the numbers in descending order: greatest, middle, and least.
```

Desktop proof of solution:

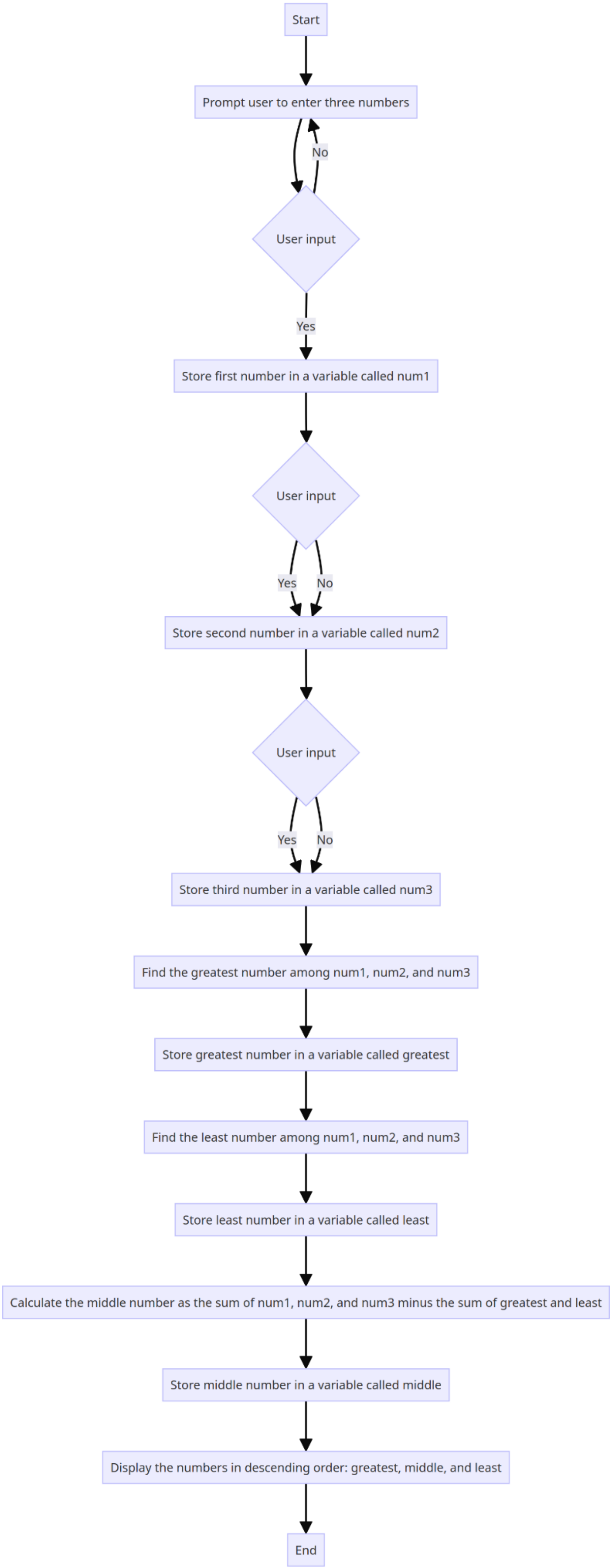
Let's say the user inputs 5, 3, and 8.

num1 = 5
num2 = 3
num3 = 8

greatest = 8
middle = 5
least = 3

The numbers in descending order are: 8, 5, 3

Flowchart:


Start
Prompt user to enter three numbers
No
User input
Yes
Store first number in a variable called num1
User input
Yes
No
Store second number in a variable called num2
User input
Yes
No
Store third number in a variable called num3
Find the greatest number among num1, num2, and num3
Store greatest number in a variable called greatest
Find the least number among num1, num2, and num3
Store least number in a variable called least
Calculate the middle number as the sum of num1, num2, and num3 minus the sum of greatest and least
Store middle number in a variable called middle
Display the numbers in descending order: greatest, middle, and least
End