ST. XAVIER'S COLLEGE

MAITIGHAR, KATHMANDU, NEPAL
Phone: 01-5321365, 01-5344636
Email: ktm@sxc.edu.np

LAB/WRITTEN ASSIGNMENT NUMBER: 9

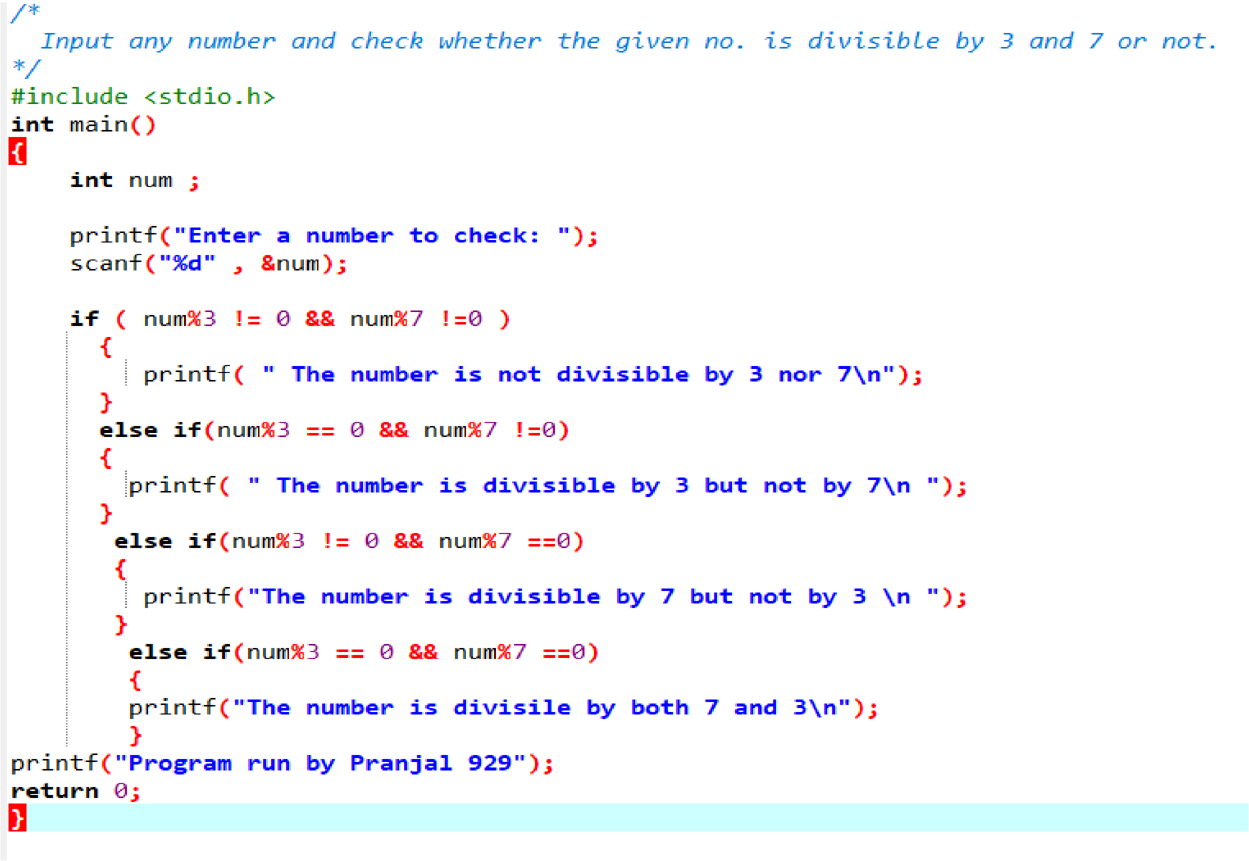
"C programming Assignment"

Submitted By	Submitted To	Signature
Name : Pranjal Khatri Roll : 929 Class : 11 Section : I	Department Of Computer Science (+2) St.Xavier's College	

Submission Date: 5 December, 2024

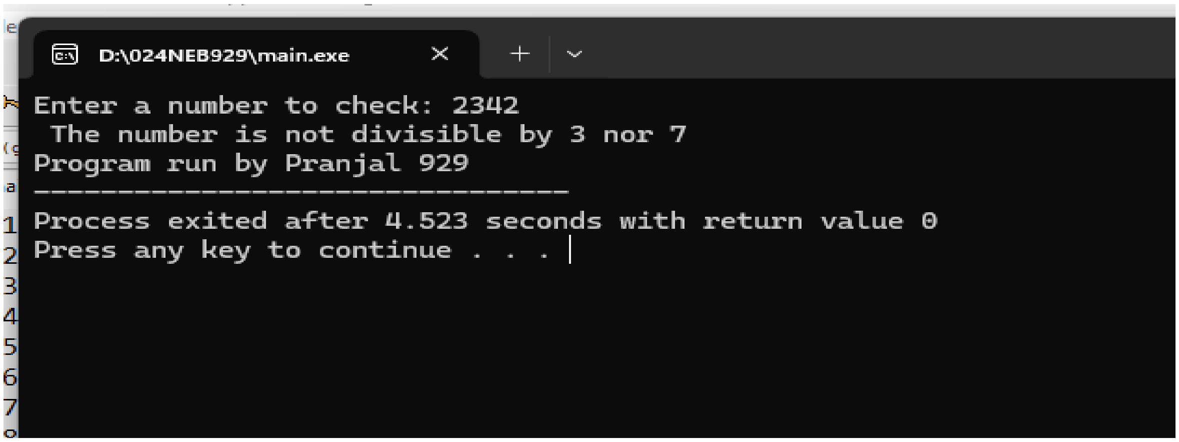
Table of Contents

Problem 1



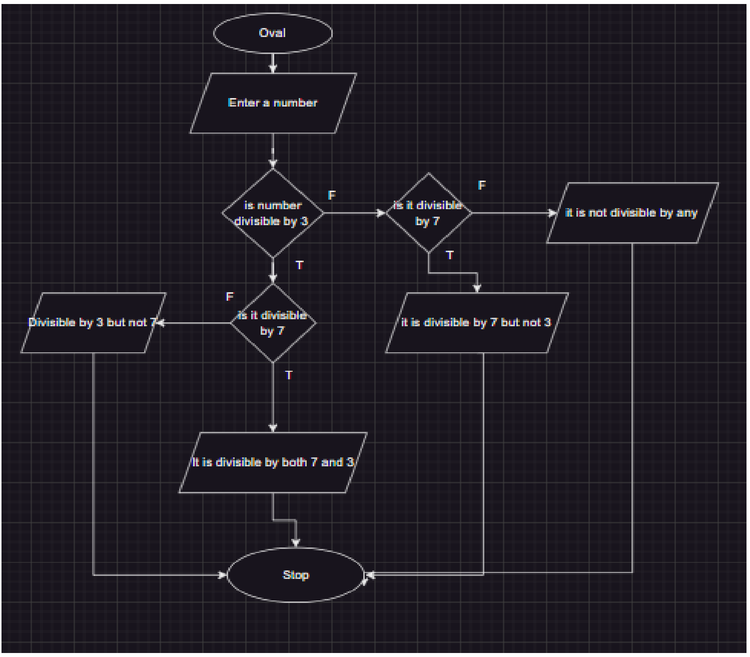
```
/*
  Input any number and check whether the given no. is divisible by 3 and 7 or not.
*/
#include <stdio.h>
int main()
{
    int num ;

    printf("Enter a number to check: ");
    scanf("%d" , &num);

    if ( num%3 != 0 && num%7 !=0 )
      {
         printf( " The number is not divisible by 3 nor 7\n");
      }
      else if(num%3 == 0 && num%7 !=0)
      {
        printf( " The number is divisible by 3 but not by 7\n ");
      }
       else if(num%3 != 0 && num%7 ==0)
       {
         printf("The number is divisible by 7 but not by 3 \n ");
       }
        else if(num%3 == 0 && num%7 ==0)
        {
        printf("The number is divisile by both 7 and 3\n");
        }
printf("Program run by Pranjal 929");
return 0;
}
```

Output:



```
Enter a number to check: 2342
 The number is not divisible by 3 nor 7
Program run by Pranjal 929
--------------------------------
Process exited after 4.523 seconds with return value 0
Press any key to continue . . .
```

Flowchart

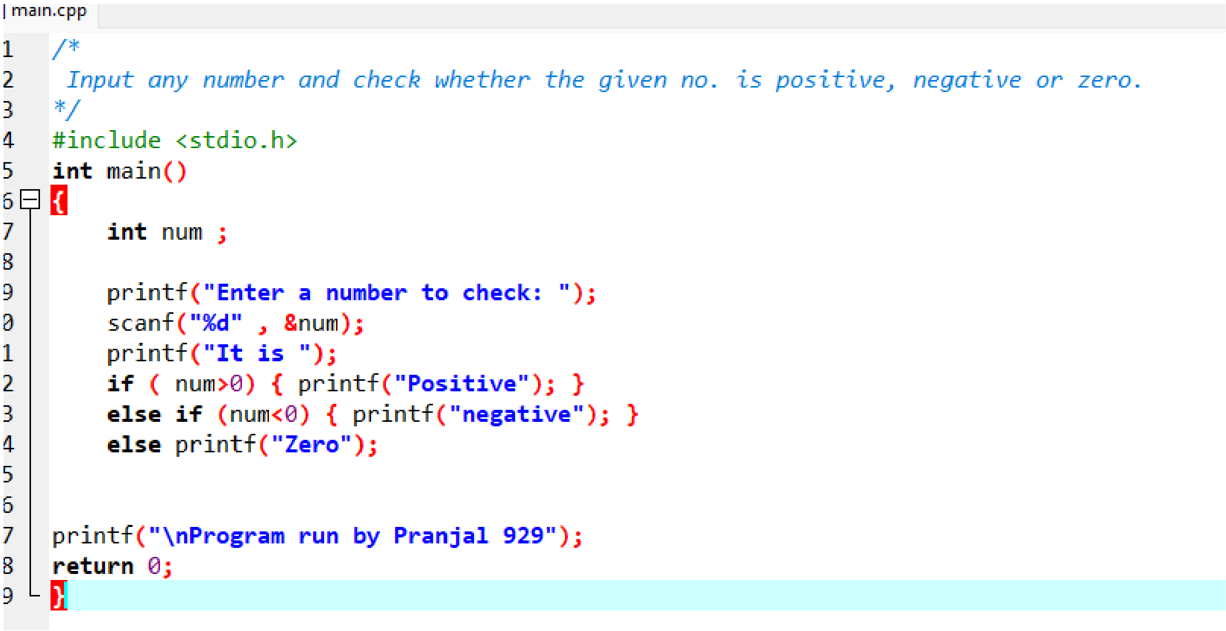


E

Algorithm

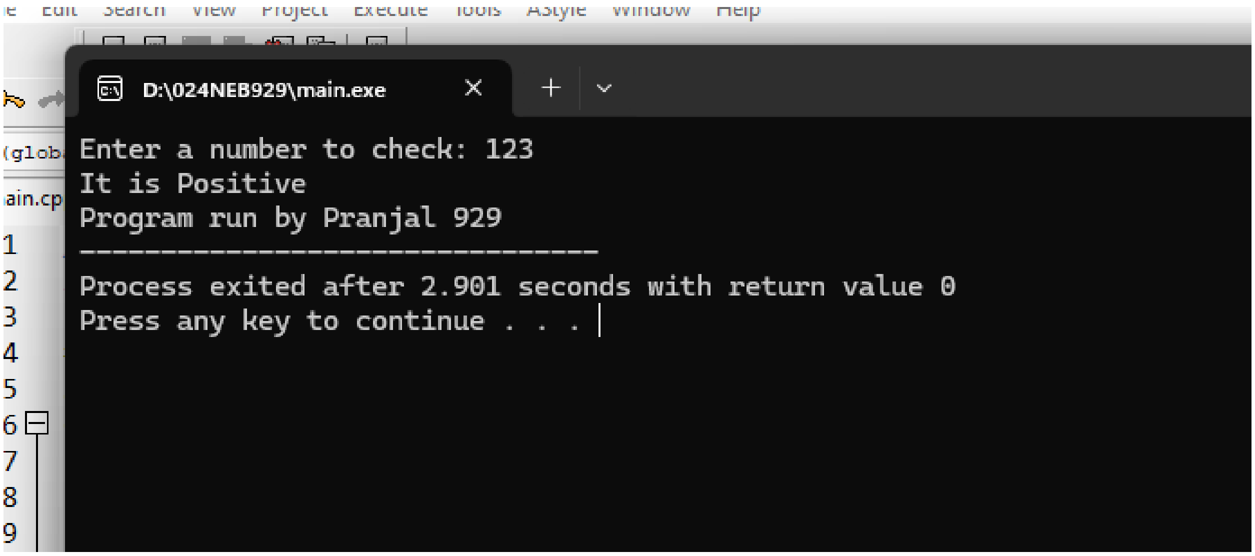
Step 1: Start.
Step 2: Input a number `num`.
Step 3: If `num` is divisible by both 3 and 7, print "Divisible by both 3 and 7."
Step 4: Else if `num` is divisible by 3 only, print "Divisible by 3 but not by 7."
Step 5: Else if `num` is divisible by 7 only, print "Divisible by 7 but not by 3."
Step 6: Else, print "Not divisible by 3 or 7."
Step 7: Print "Program run by Pranjal 929."
Step 8: Stop.

Problem 2



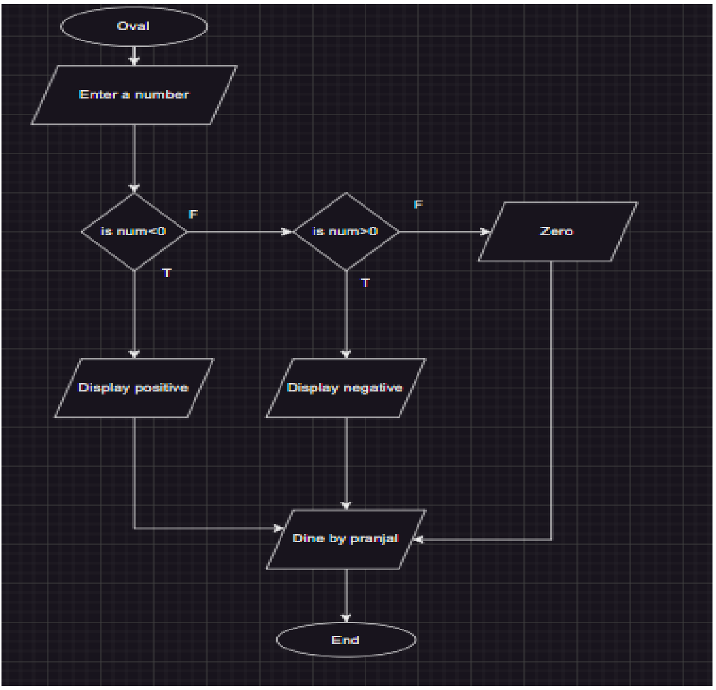
```c
/*
 Input any number and check whether the given no. is positive, negative or zero.
*/
#include <stdio.h>
int main()
{
    int num ;

    printf("Enter a number to check: ");
    scanf("%d" , &num);
    printf("It is ");
    if ( num>0) { printf("Positive"); }
    else if (num<0) { printf("negative"); }
    else printf("Zero");


printf("\nProgram run by Pranjal 929");
return 0;
}
```

Output :



```
Enter a number to check: 123
It is Positive
Program run by Pranjal 929
--------------------------------
Process exited after 2.901 seconds with return value 0
Press any key to continue . . .
```

Flowchart

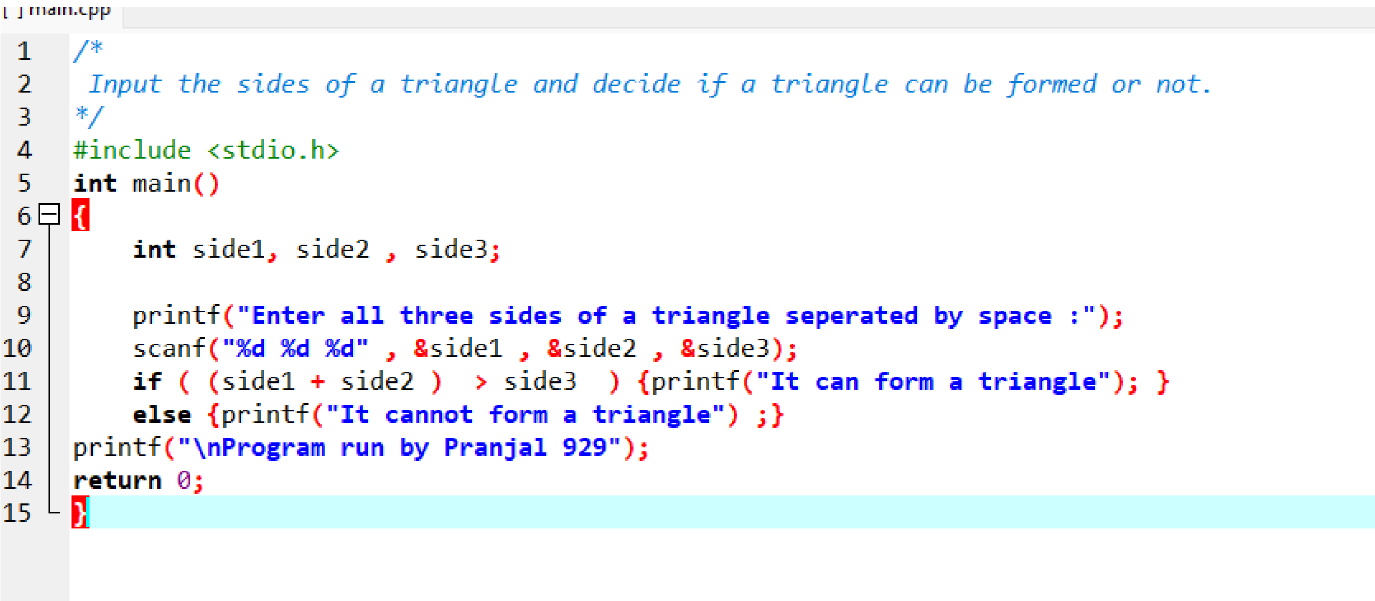


E

Algorithm

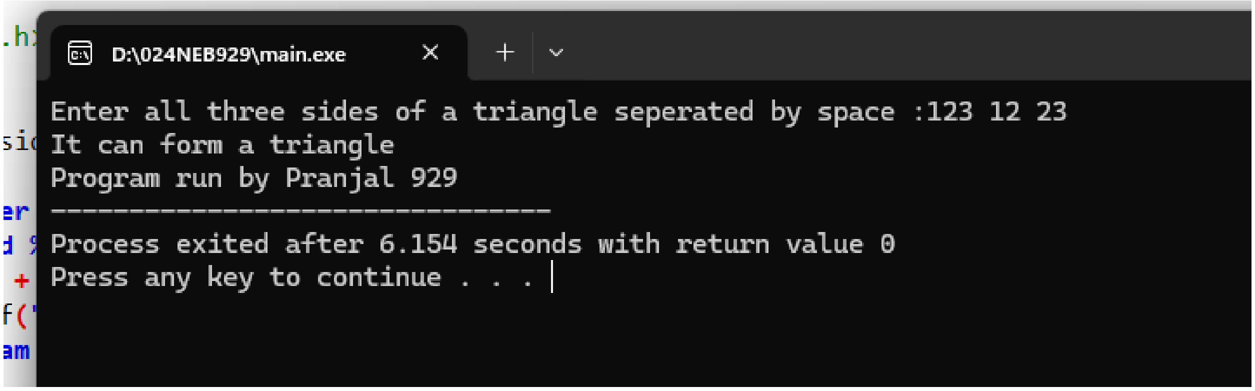
Step 1: Start.
Step 2: Input a number `num`.
Step 3: Print "It is ".
Step 4: If `num > 0`, print "Positive".
Step 5: Else if `num < 0`, print "Negative".
Step 6: Else, print "Zero".
Step 7: Print "Program run by Pranjal 929".
Step 8: End.

Problem 3



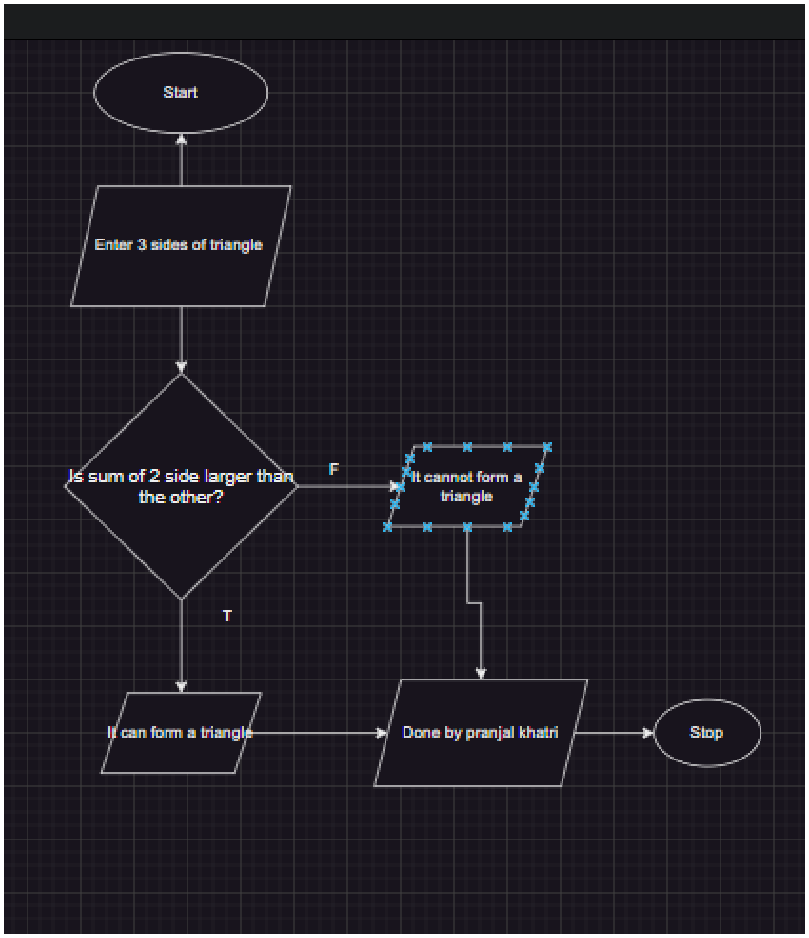
```c
/*
 Input the sides of a triangle and decide if a triangle can be formed or not.
*/
#include <stdio.h>
int main()
{
    int side1, side2 , side3;

    printf("Enter all three sides of a triangle seperated by space :");
    scanf("%d %d %d" , &side1 , &side2 , &side3);
    if ( (side1 + side2 )  > side3  ) {printf("It can form a triangle"); }
    else {printf("It cannot form a triangle") ;}
printf("\nProgram run by Pranjal 929");
return 0;
}
```

Output



```
Enter all three sides of a triangle seperated by space :123 12 23
It can form a triangle
Program run by Pranjal 929
--------------------------------
Process exited after 6.154 seconds with return value 0
Press any key to continue . . .
```

Flowchart

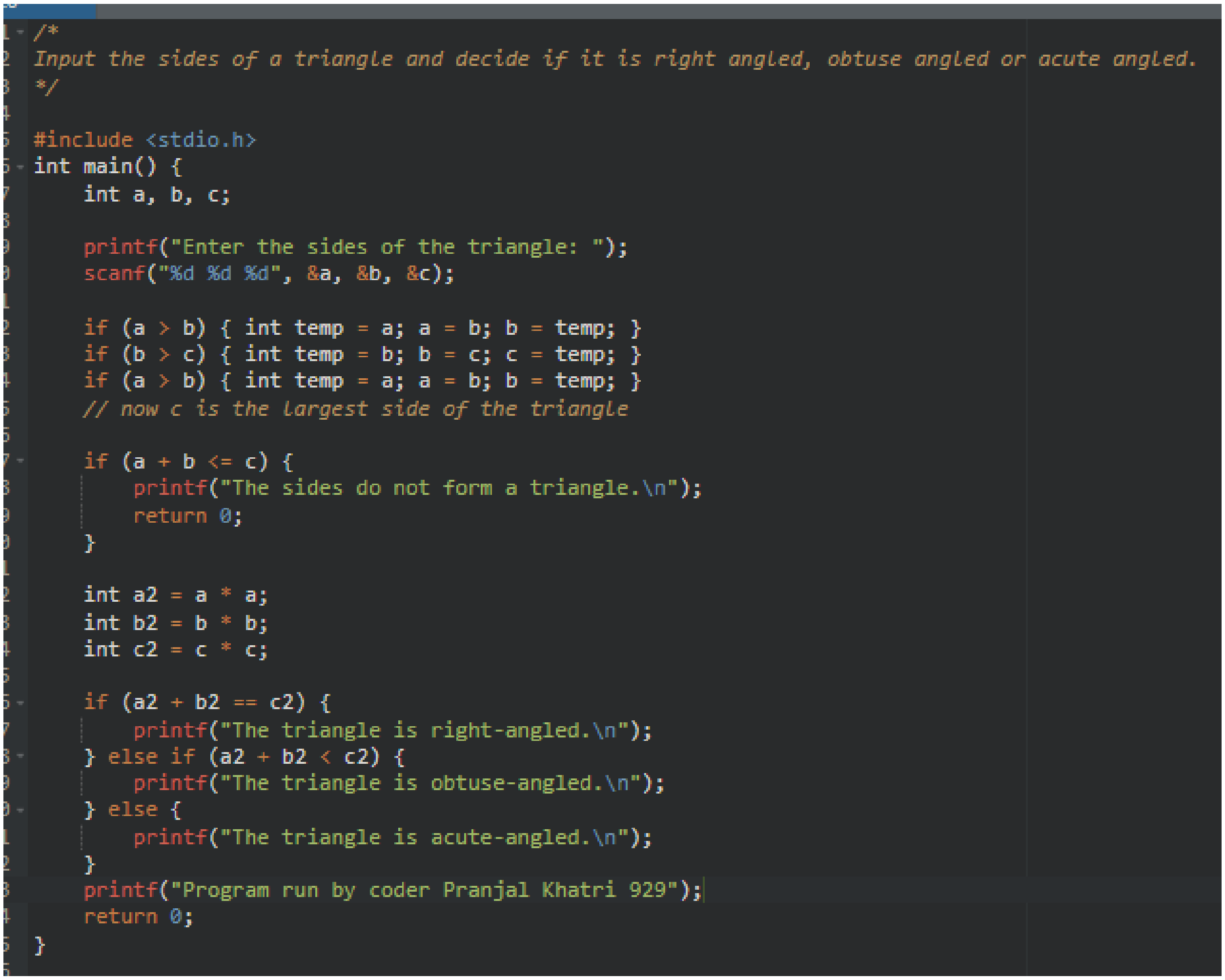

Start
Enter 3 sides of triangle
Is sum of 2 side larger than the other?
F
It cannot form a triangle
T
It can form a triangle
Done by pranjal khatri
Stop

E

Algorithm

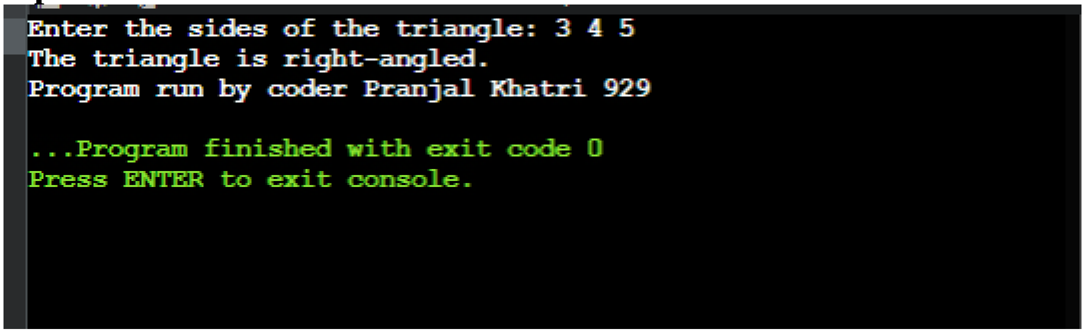
Step 1: Start.
Step 2: Input three sides of a triangle: `side1`, `side2`, and `side3`.
Step 3: If the sum of `side1 + side2` is greater than `side3`, print "It can form a triangle".
Step 4: Else, print "It cannot form a triangle".
Step 5: Print "Program run by Pranjal 929".
Step 6: End.

Problem 4



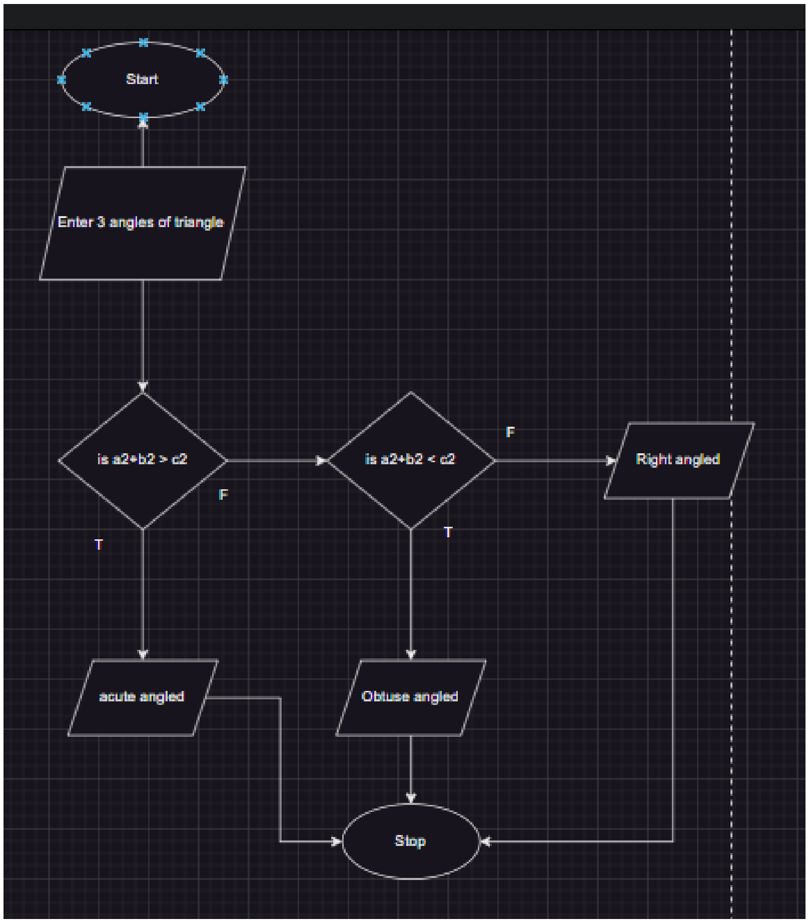
```c
/*
Input the sides of a triangle and decide if it is right angled, obtuse angled or acute angled.
*/

#include <stdio.h>
int main() {
    int a, b, c;

    printf("Enter the sides of the triangle: ");
    scanf("%d %d %d", &a, &b, &c);

    if (a > b) { int temp = a; a = b; b = temp; }
    if (b > c) { int temp = b; b = c; c = temp; }
    if (a > b) { int temp = a; a = b; b = temp; }
    // now c is the largest side of the triangle

    if (a + b <= c) {
        printf("The sides do not form a triangle.\n");
        return 0;
    }

    int a2 = a * a;
    int b2 = b * b;
    int c2 = c * c;

    if (a2 + b2 == c2) {
        printf("The triangle is right-angled.\n");
    } else if (a2 + b2 < c2) {
        printf("The triangle is obtuse-angled.\n");
    } else {
        printf("The triangle is acute-angled.\n");
    }
    printf("Program run by coder Pranjal Khatri 929");
    return 0;
}
```

Output



```
Enter the sides of the triangle: 3 4 5
The triangle is right-angled.
Program run by coder Pranjal Khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

Flowchart



Algorithm

Step 1: Start.

Step 2: Input three sides of a triangle: `a`, `b`, and `c`.

Step 3: Arrange the sides such that `c` is the largest side using swapping if necessary.

Step 4: If the sum of `a + b` is less than or equal to `c`, print "The sides do not form a triangle" and go to Step 8.

Step 5: Calculate the squares of the sides: `a2 = a * a`, `b2 = b * b`, and `c2 = c * c`.

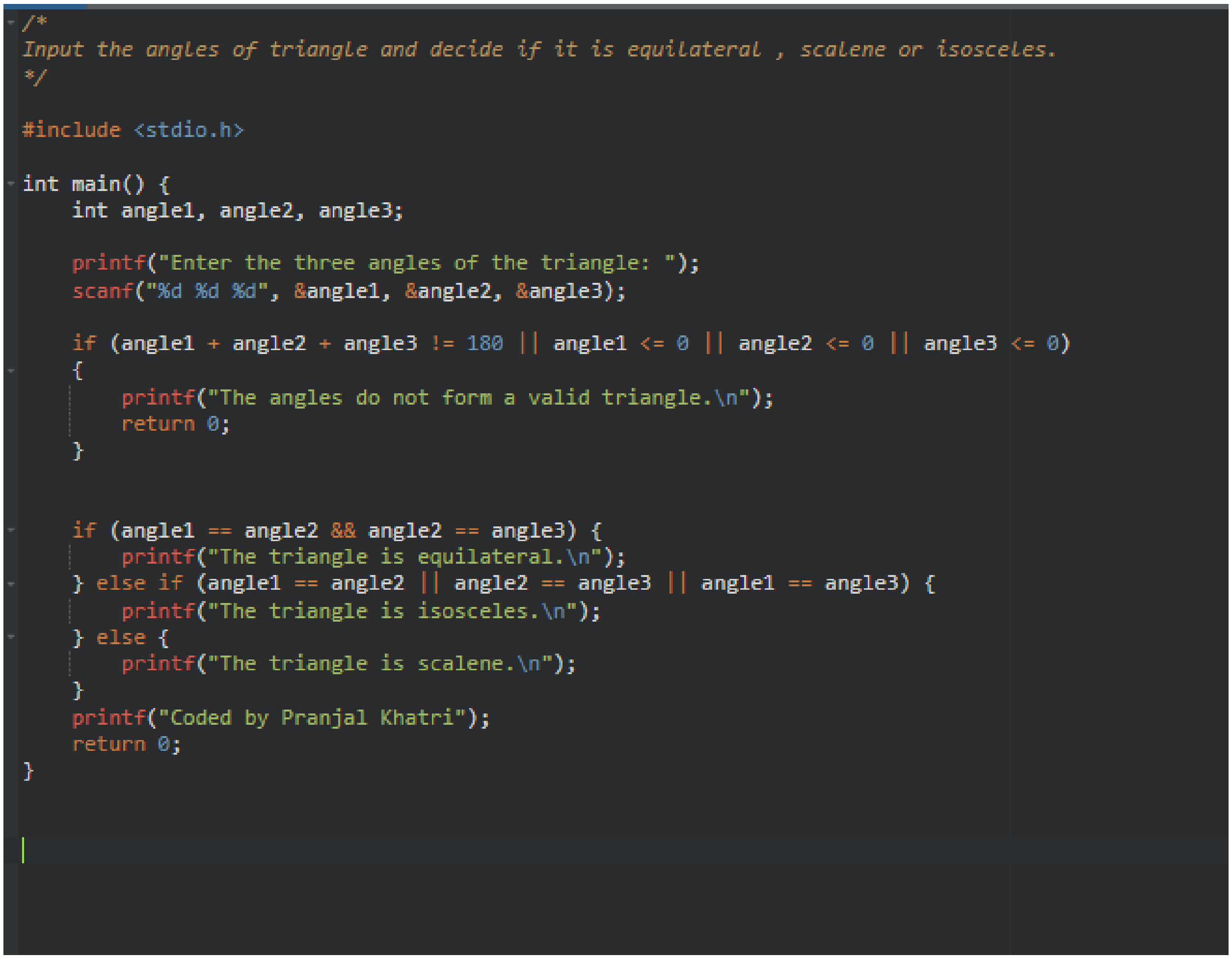
Step 6: If `a2 + b2 == c2`, print "The triangle is right-angled".

Step 7: If `a2 + b2 < c2`, print "The triangle is obtuse-angled"; otherwise, print "The triangle is acute-angled".

Step 8: Print "Program run by Pranjal 929".

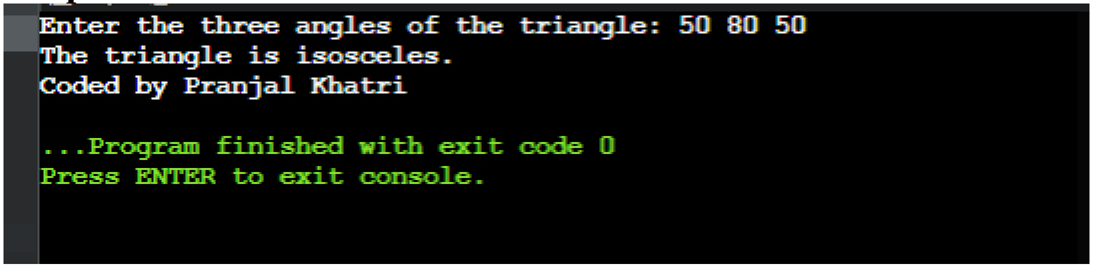
Step 9: End.

Problem 5



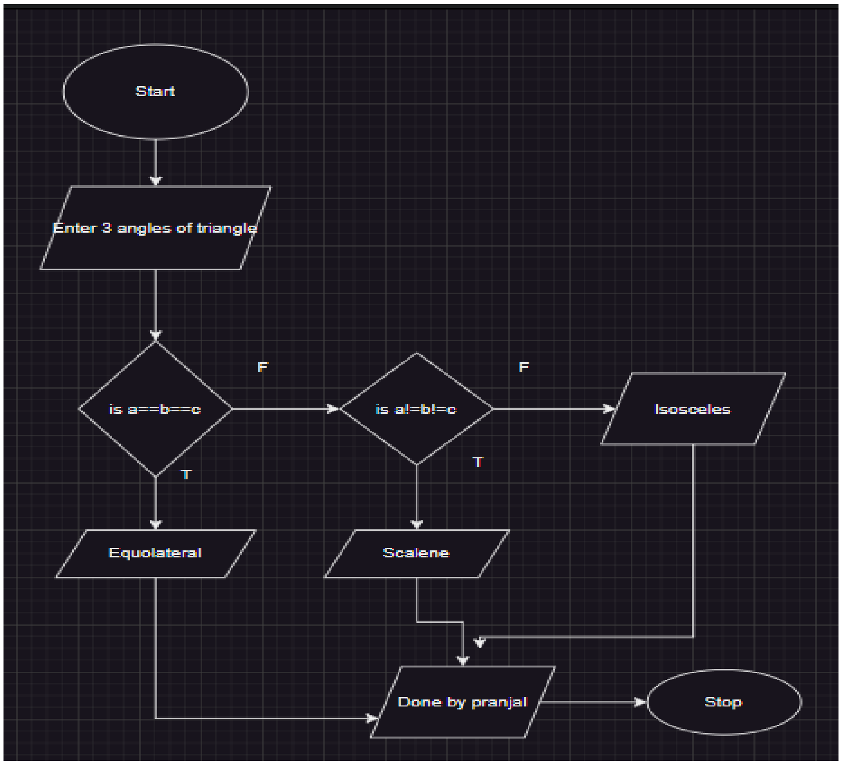
```
/*
Input the angles of triangle and decide if it is equilateral , scalene or isosceles.
*/

#include <stdio.h>

int main() {
    int angle1, angle2, angle3;

    printf("Enter the three angles of the triangle: ");
    scanf("%d %d %d", &angle1, &angle2, &angle3);

    if (angle1 + angle2 + angle3 != 180 || angle1 <= 0 || angle2 <= 0 || angle3 <= 0)
    {
        printf("The angles do not form a valid triangle.\n");
        return 0;
    }


    if (angle1 == angle2 && angle2 == angle3) {
        printf("The triangle is equilateral.\n");
    } else if (angle1 == angle2 || angle2 == angle3 || angle1 == angle3) {
        printf("The triangle is isosceles.\n");
    } else {
        printf("The triangle is scalene.\n");
    }
    printf("Coded by Pranjal Khatri");
    return 0;
}
```

Output



```
Enter the three angles of the triangle: 50 80 50
The triangle is isosceles.
Coded by Pranjal Khatri

...Program finished with exit code 0
Press ENTER to exit console.
```

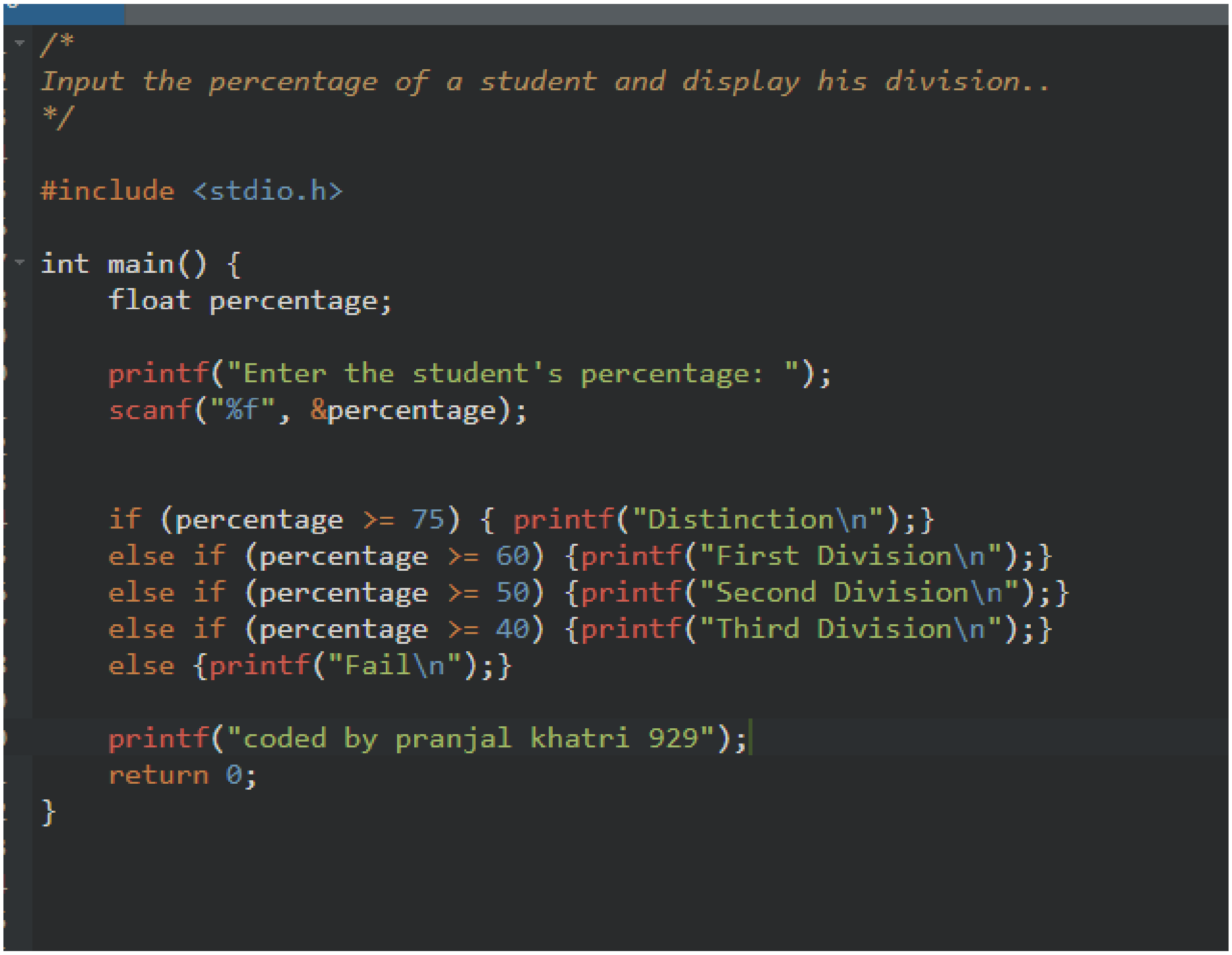
Flowchart



Algorithm

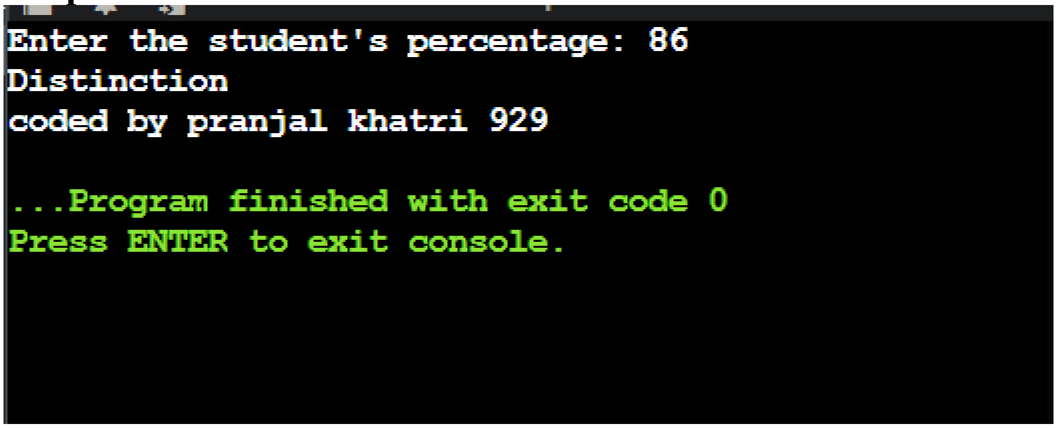
Step 1: Start.
Step 2: Input three angles of a triangle: `angle1`, `angle2`, and `angle3`.
Step 3: Check if the sum of `angle1 + angle2 + angle3` is not equal to 180 or if any angle is less than or equal to 0.
Step 4: If the condition in Step 3 is true, print "The angles do not form a valid triangle" and go to Step 8.
Step 5: If all three angles are equal, print "The triangle is equilateral".
Step 6: Else, if any two angles are equal, print "The triangle is isosceles".
Step 7: Otherwise, print "The triangle is scalene".
Step 8: Print "Coded by Pranjal Khatri".
Step 9: End.

Problem 6



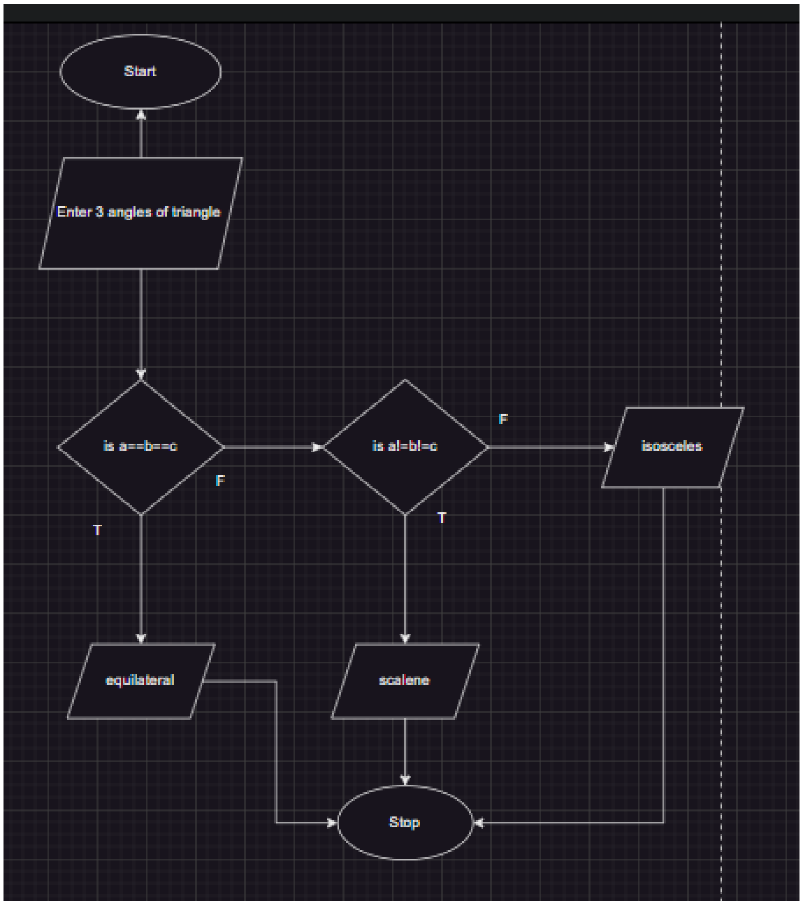
```c
/*
Input the percentage of a student and display his division..
*/

#include <stdio.h>

int main() {
    float percentage;

    printf("Enter the student's percentage: ");
    scanf("%f", &percentage);


    if (percentage >= 75) { printf("Distinction\n");}
    else if (percentage >= 60) {printf("First Division\n");}
    else if (percentage >= 50) {printf("Second Division\n");}
    else if (percentage >= 40) {printf("Third Division\n");}
    else {printf("Fail\n");}

    printf("coded by pranjal khatri 929");
    return 0;
}
```

Output



```
Enter the student's percentage: 86
Distinction
coded by pranjal khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

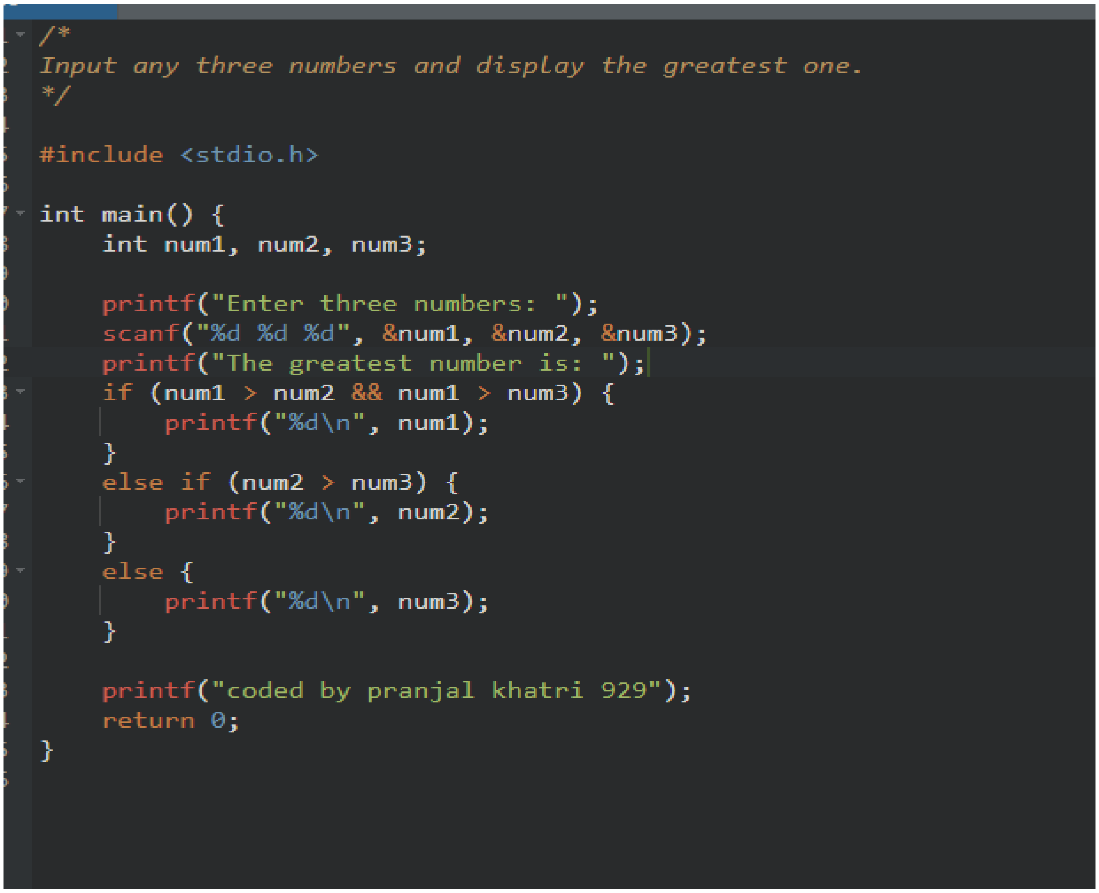
Flowchart



Algorithm

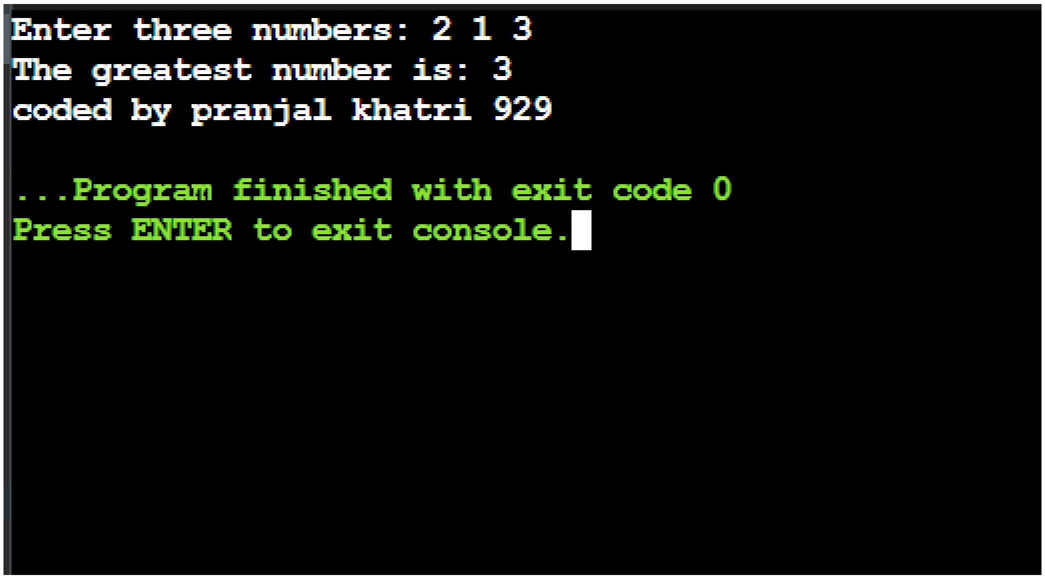
Step 1: Start.
Step 2: Input the percentage of the student.
Step 3: If the percentage is greater than or equal to 75, print "Distinction".
Step 4: Else if the percentage is greater than or equal to 60, print "First Division".
Step 5: Else if the percentage is greater than or equal to 50, print "Second Division".
Step 6: Else if the percentage is greater than or equal to 40, print "Third Division".
Step 7: Else, print "Fail".
Step 8: Print "Program run by Pranjal 929".
Step 9: End.

Problem 7



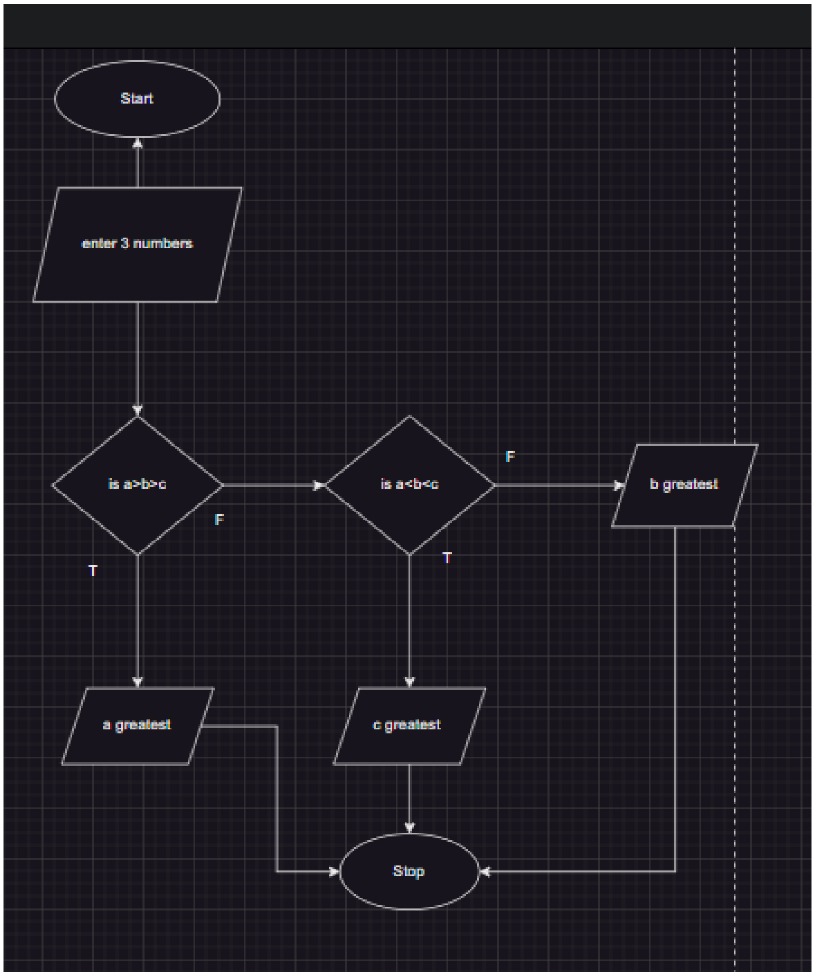
```c
/*
Input any three numbers and display the greatest one.
*/

#include <stdio.h>

int main() {
    int num1, num2, num3;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);
    printf("The greatest number is: ");
    if (num1 > num2 && num1 > num3) {
        printf("%d\n", num1);
    }
    else if (num2 > num3) {
        printf("%d\n", num2);
    }
    else {
        printf("%d\n", num3);
    }

    printf("coded by pranjal khatri 929");
    return 0;
}
```

Output



```
Enter three numbers: 2 1 3
The greatest number is: 3
coded by pranjal khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

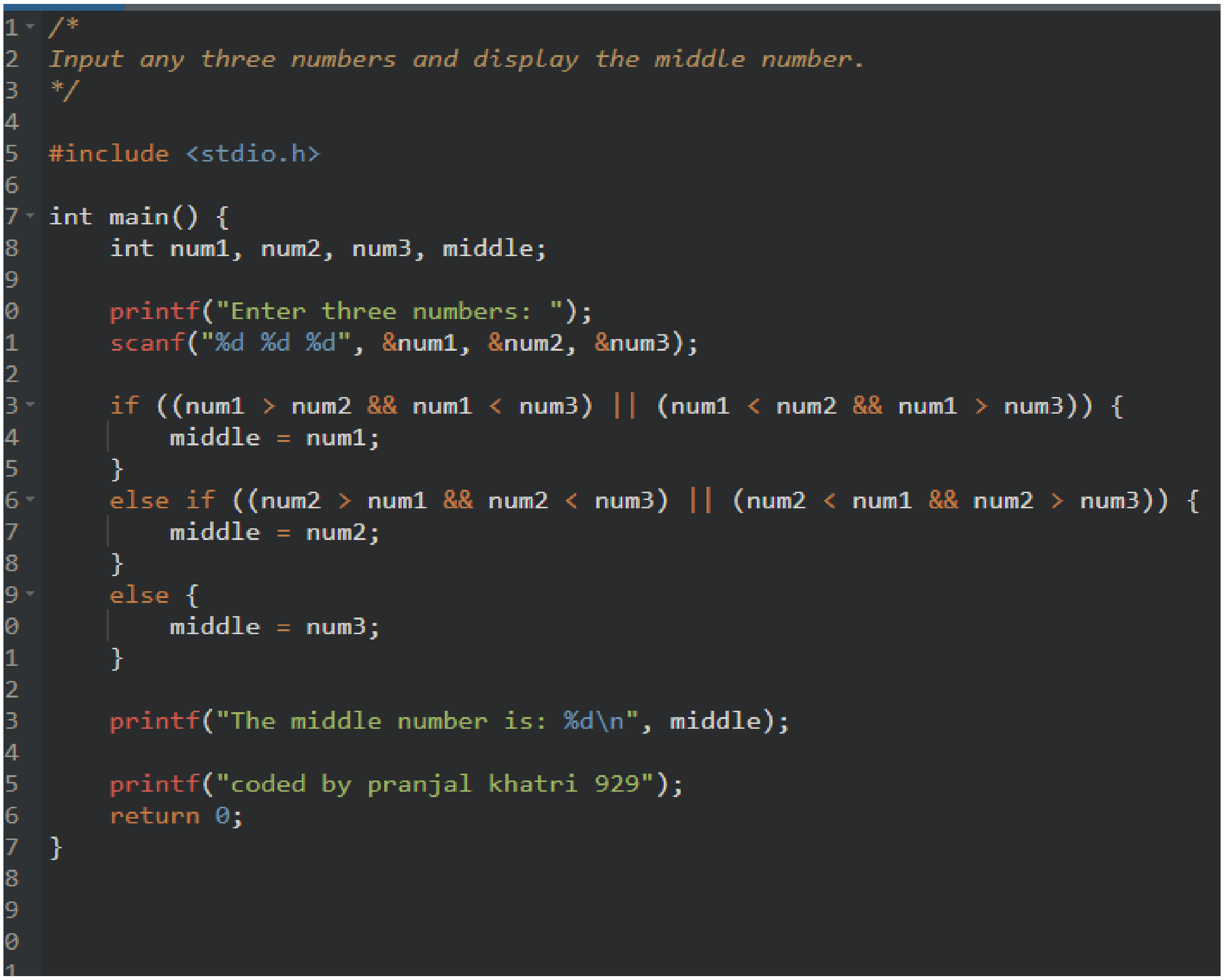
Flowchart



Algorithm

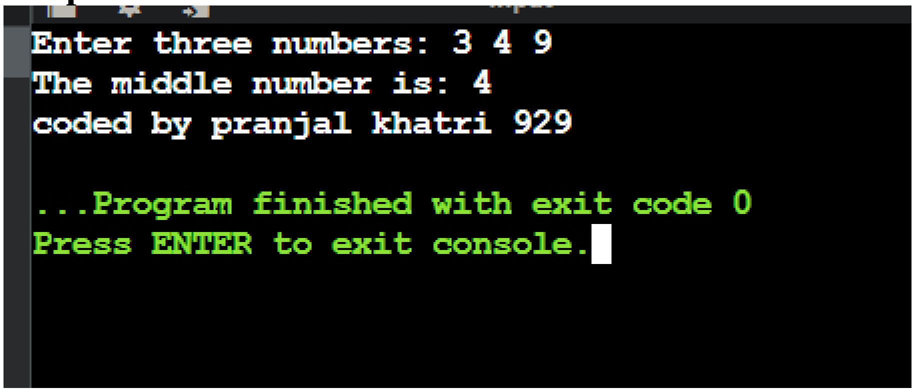
Step 1: Start.
Step 2: Input three numbers: `num1`, `num2`, and `num3`.
Step 3: If `num1` is greater than `num2` and `num3`, print `num1` as the greatest number.
Step 4: Else if `num2` is greater than `num3`, print `num2` as the greatest number.
Step 5: Else, print `num3` as the greatest number.
Step 6: Print "coded by Pranjal Khatri 929".
Step 7: End.

Problem 8



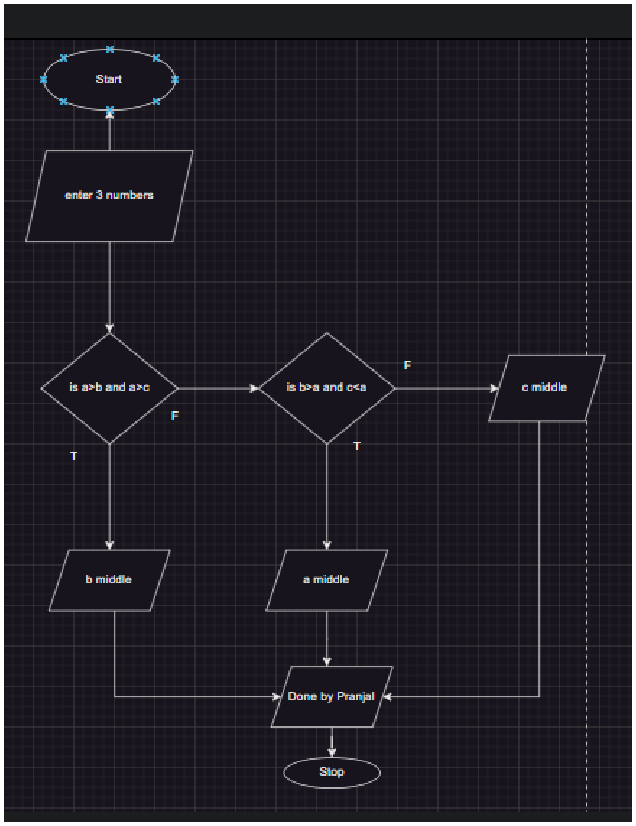
```c
/*
Input any three numbers and display the middle number.
*/

#include <stdio.h>

int main() {
    int num1, num2, num3, middle;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    if ((num1 > num2 && num1 < num3) || (num1 < num2 && num1 > num3)) {
        middle = num1;
    }
    else if ((num2 > num1 && num2 < num3) || (num2 < num1 && num2 > num3)) {
        middle = num2;
    }
    else {
        middle = num3;
    }

    printf("The middle number is: %d\n", middle);

    printf("coded by pranjal khatri 929");
    return 0;
}
```

Output



```
Enter three numbers: 3 4 9
The middle number is: 4
coded by pranjal khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

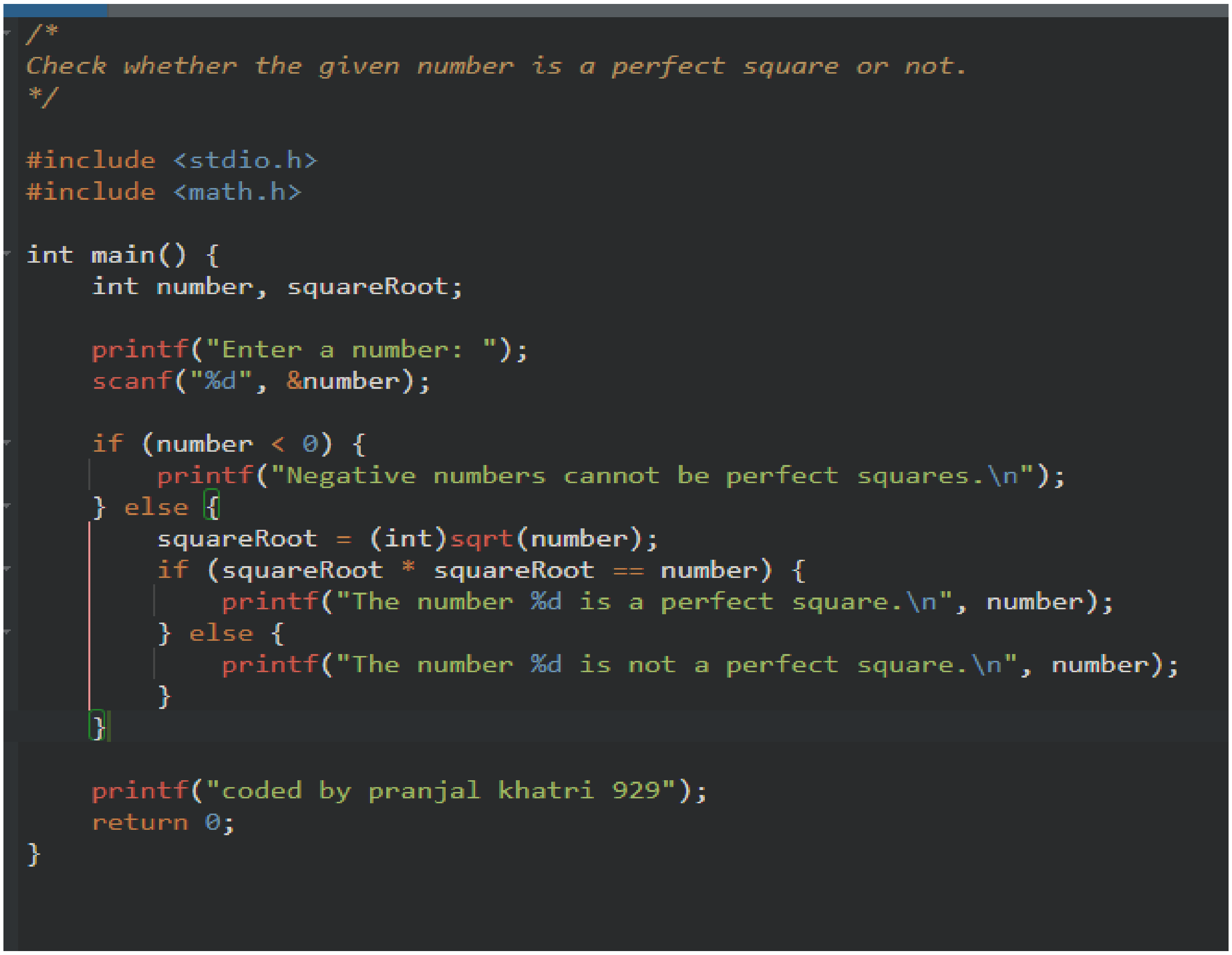
Flowchart



Algorithm

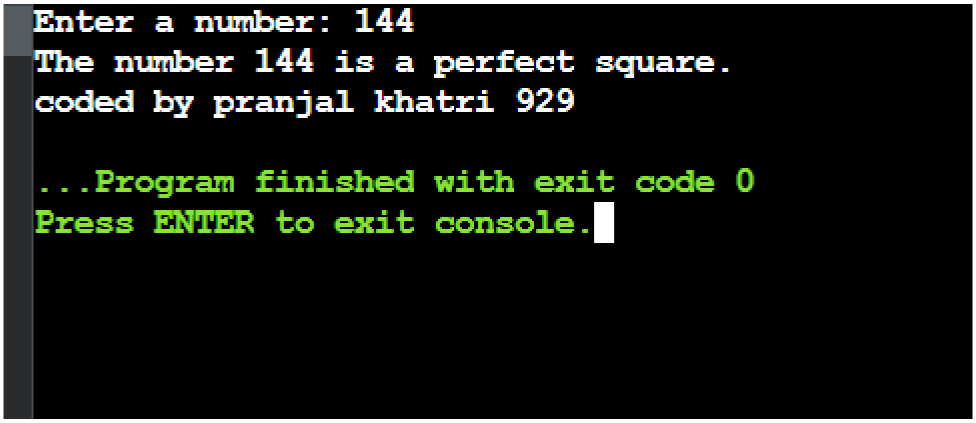
Step 1: Start.
Step 2: Input three numbers: `num1`, `num2`, and `num3`.
Step 3: If `num1` is greater than `num2` and less than `num3`, or less than `num2` and greater than `num3`, set `middle = num1`.
Step 4: Else if `num2` is greater than `num1` and less than `num3`, or less than `num1` and greater than `num3`, set `middle = num2`.
Step 5: Else, set `middle = num3`.
Step 6: Print the middle number.
Step 7: Print "coded by Pranjal Khatri 929".
Step 8: End.

Problem 9



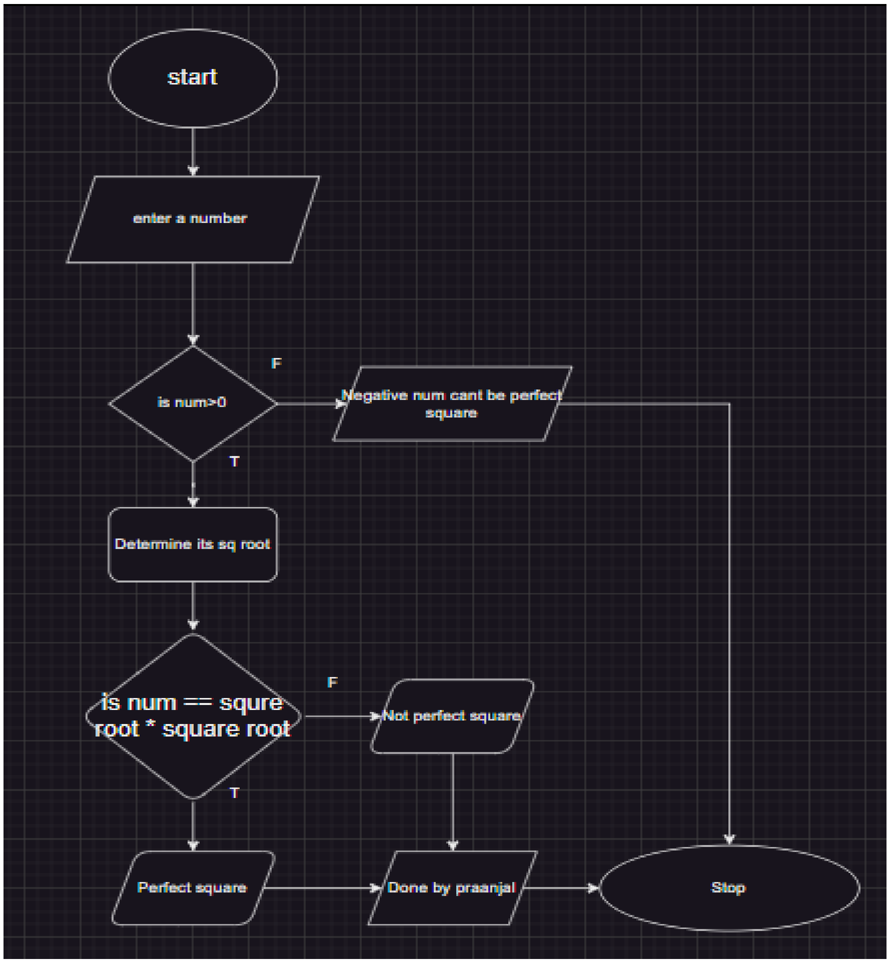
```c
/*
Check whether the given number is a perfect square or not.
*/

#include <stdio.h>
#include <math.h>

int main() {
    int number, squareRoot;

    printf("Enter a number: ");
    scanf("%d", &number);

    if (number < 0) {
        printf("Negative numbers cannot be perfect squares.\n");
    } else {
        squareRoot = (int)sqrt(number);
        if (squareRoot * squareRoot == number) {
            printf("The number %d is a perfect square.\n", number);
        } else {
            printf("The number %d is not a perfect square.\n", number);
        }
    }

    printf("coded by pranjal khatri 929");
    return 0;
}
```

Output



```
Enter a number: 144
The number 144 is a perfect square.
coded by pranjal khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

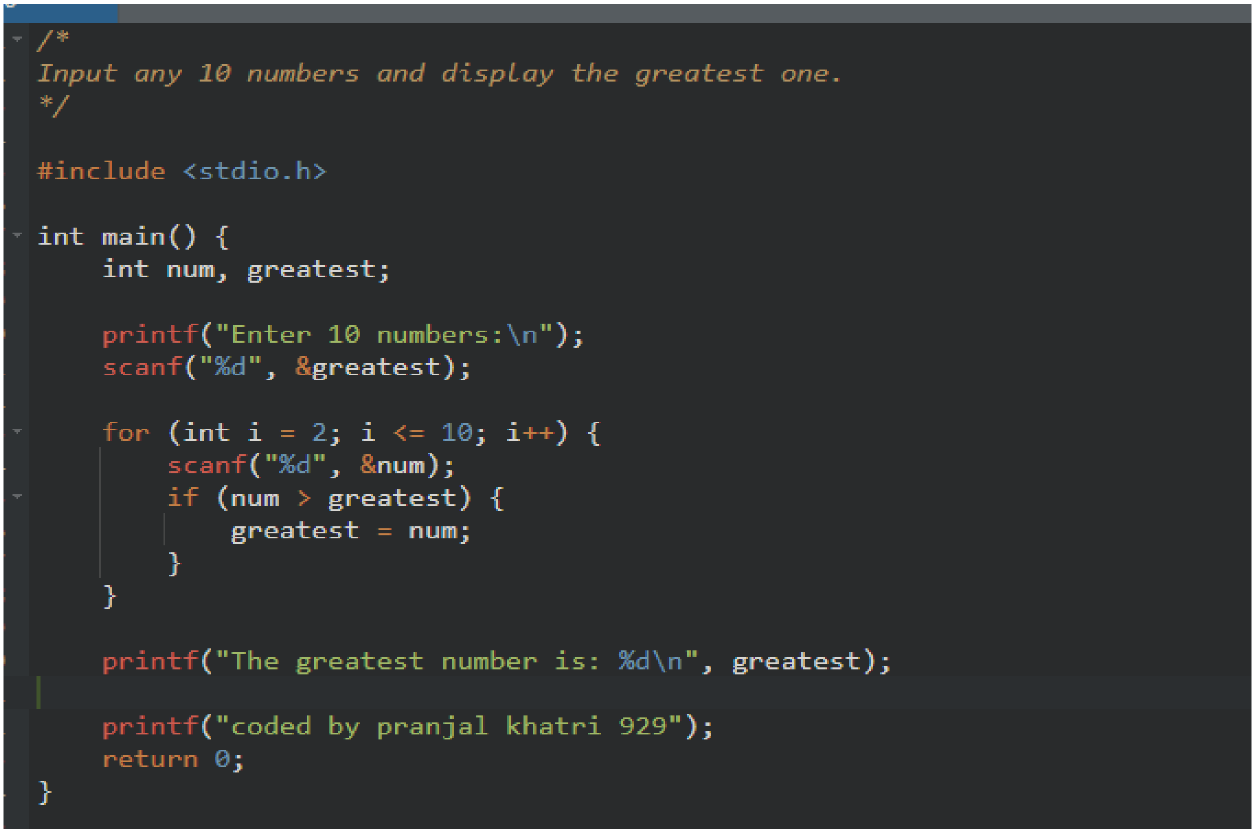
Flowchart



Algorithm

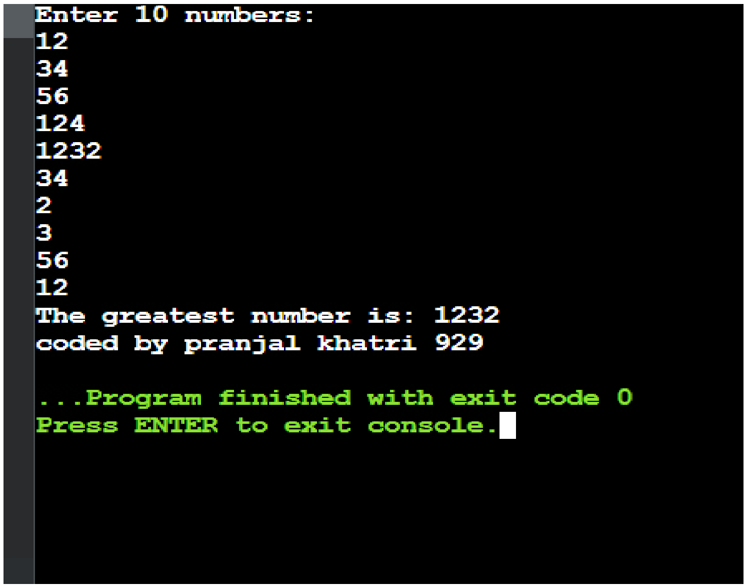
Step 1: Start.
Step 2: Input a number `number`.
Step 3: If the number is negative, print "Negative numbers cannot be perfect squares."
Step 4: Otherwise, calculate the square root of the number and store it in `squareRoot`.
Step 5: If `squareRoot * squareRoot == number`, print "The number is a perfect square."
Step 6: Else, print "The number is not a perfect square."
Step 7: Print "coded by pranjal khatri 929."
Step 8: Stop.

Problem 10



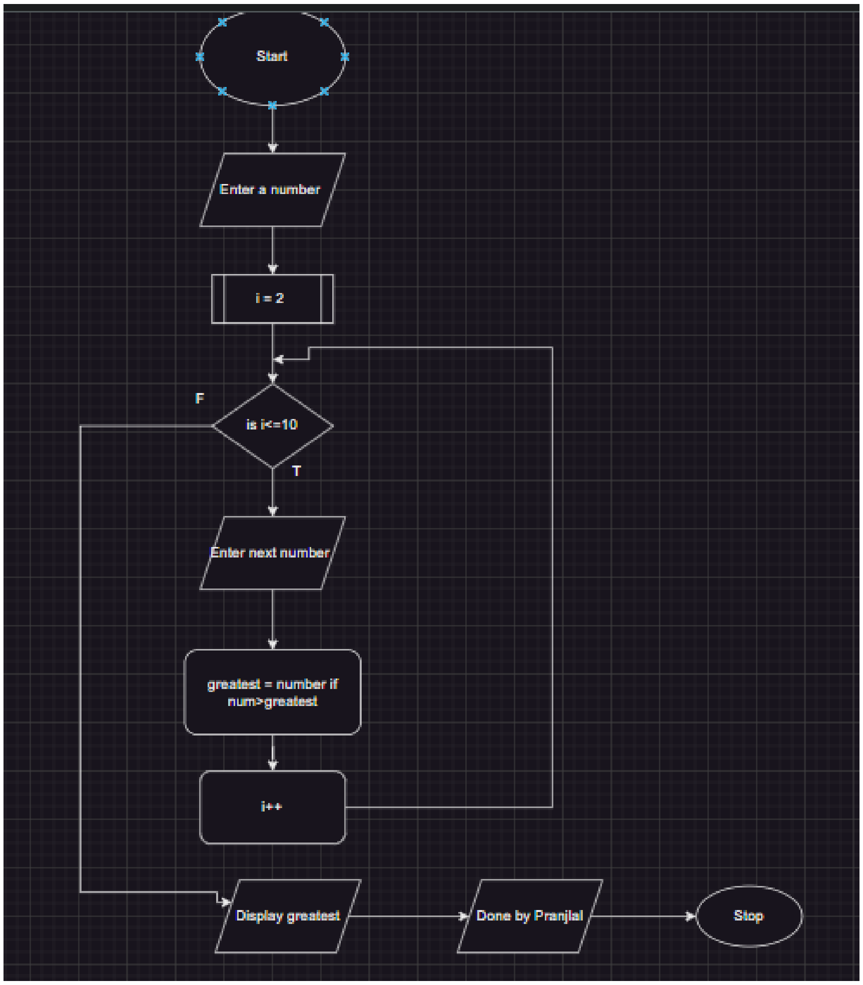
```c
/*
Input any 10 numbers and display the greatest one.
*/

#include <stdio.h>

int main() {
    int num, greatest;

    printf("Enter 10 numbers:\n");
    scanf("%d", &greatest);

    for (int i = 2; i <= 10; i++) {
        scanf("%d", &num);
        if (num > greatest) {
            greatest = num;
        }
    }

    printf("The greatest number is: %d\n", greatest);

    printf("coded by pranjal khatri 929");
    return 0;
}
```

Output



```
Enter 10 numbers:
12
34
56
124
1232
34
2
3
56
12
The greatest number is: 1232
coded by pranjal khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

Flowchart



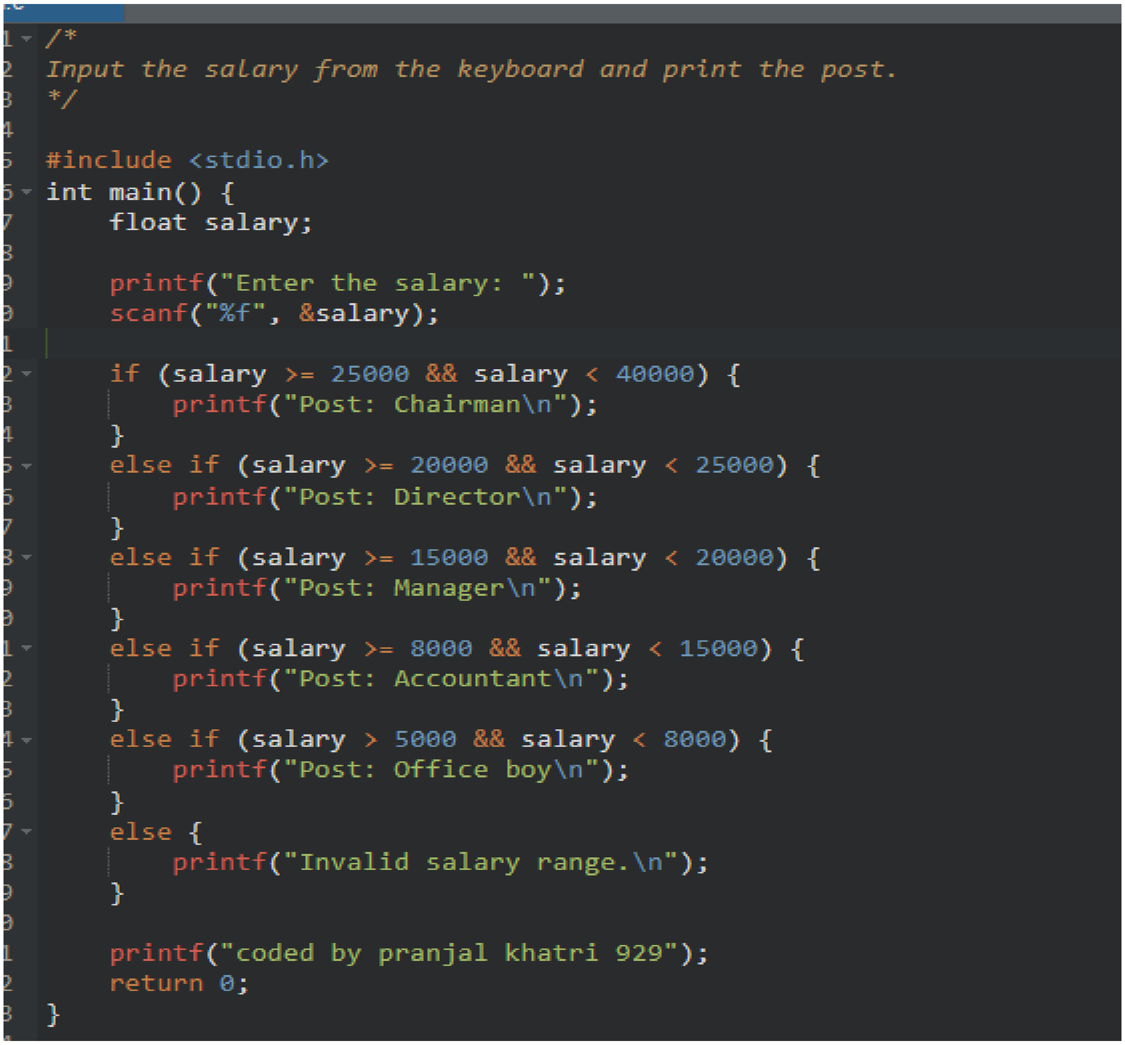
Algorithm

Step 1: Start.
Step 2: Input the first number and assign it to greatest.
Step 3: Loop from 2 to 10 and do the following:

- Input a number num.
- If num is greater than greatest, set greatest = num.

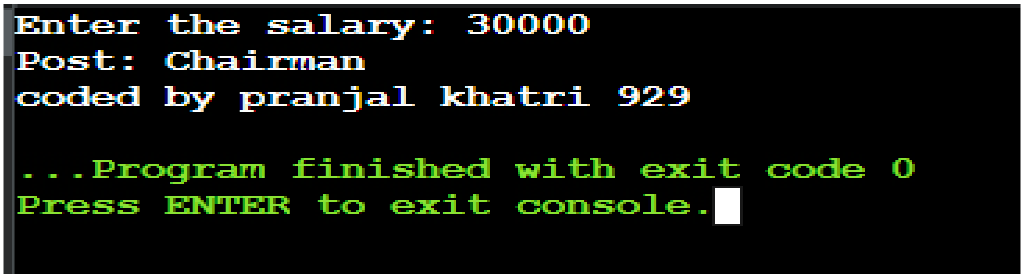
Step 4: After the loop, print the greatest number.
Step 5: Print "coded by Pranjal Khatri 929".
Step 6: End.

Problem 11



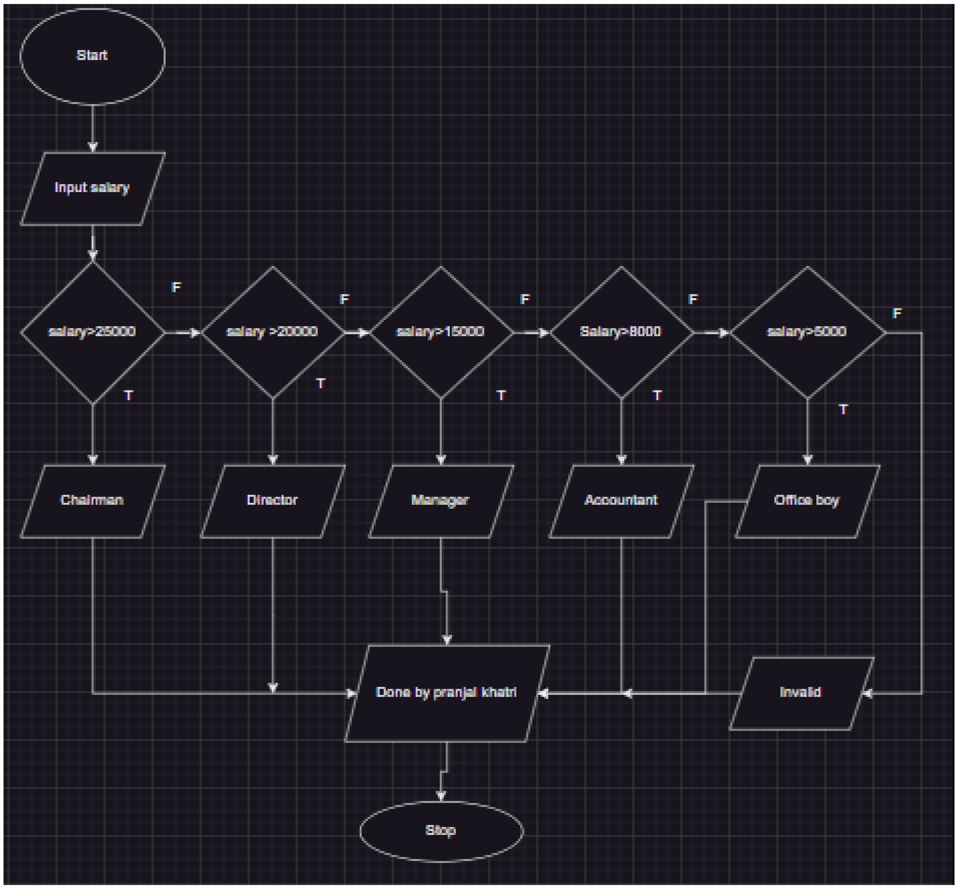
```
/*
Input the salary from the keyboard and print the post.
*/

#include <stdio.h>
int main() {
    float salary;

    printf("Enter the salary: ");
    scanf("%f", &salary);

    if (salary >= 25000 && salary < 40000) {
        printf("Post: Chairman\n");
    }
    else if (salary >= 20000 && salary < 25000) {
        printf("Post: Director\n");
    }
    else if (salary >= 15000 && salary < 20000) {
        printf("Post: Manager\n");
    }
    else if (salary >= 8000 && salary < 15000) {
        printf("Post: Accountant\n");
    }
    else if (salary > 5000 && salary < 8000) {
        printf("Post: Office boy\n");
    }
    else {
        printf("Invalid salary range.\n");
    }

    printf("coded by pranjal khatri 929");
    return 0;
}
```

Output



```
Enter the salary: 30000
Post: Chairman
coded by pranjal khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

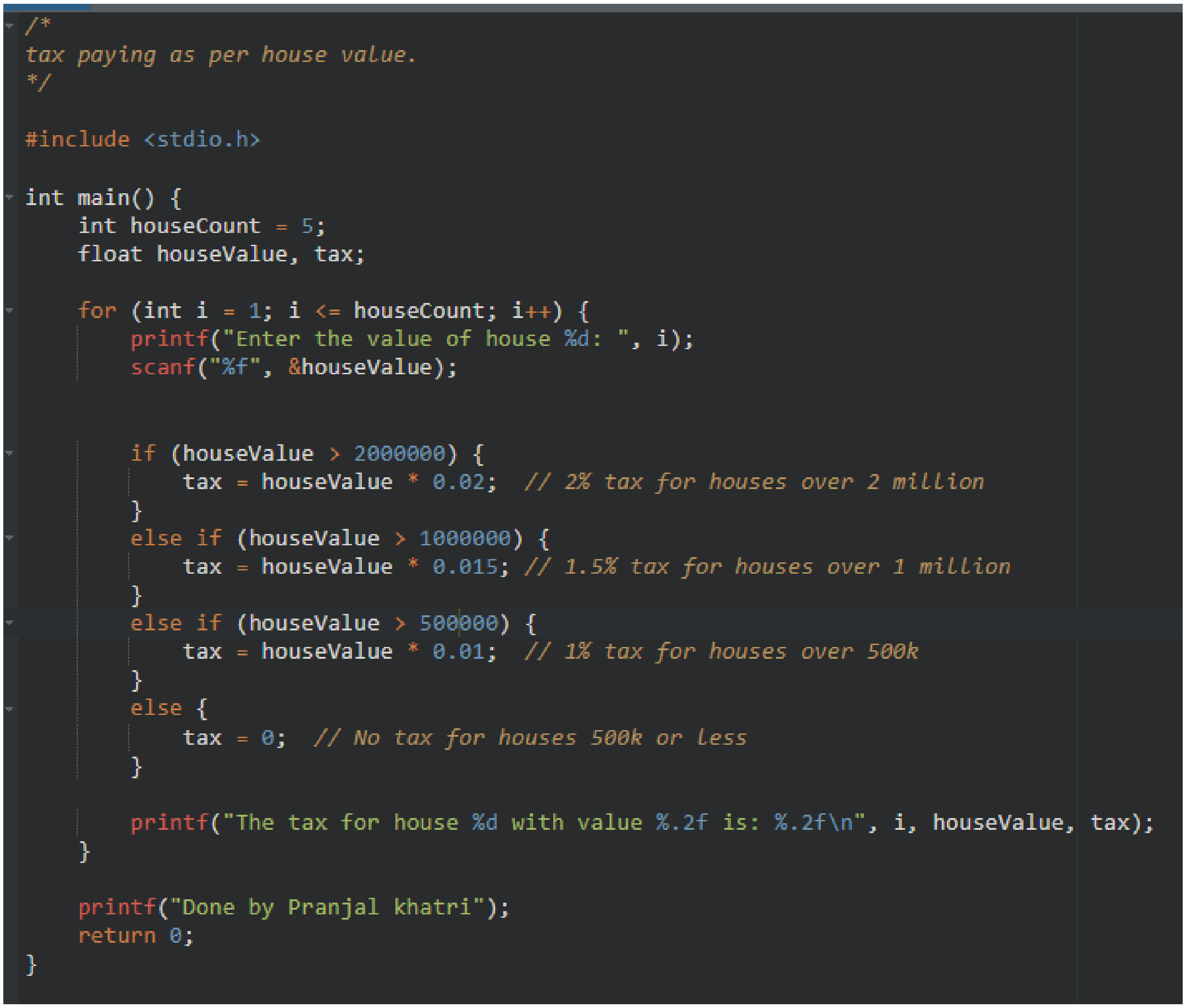
Flowchart



Algorithm

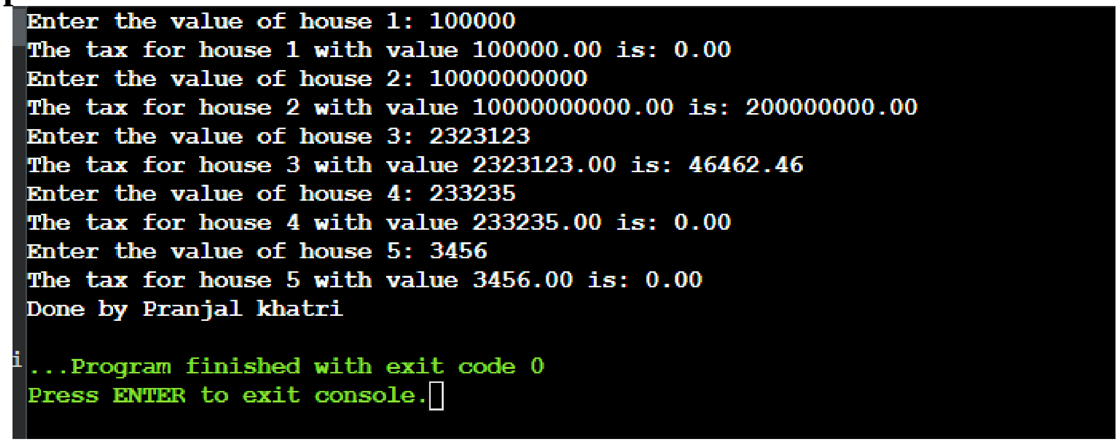
Step 1: Start.
Step 2: Input the salary.
Step 3: If salary is between 25000 and 40000, print "Post: Chairman".
Step 4: Else if salary is between 20000 and 25000, print "Post: Director".
Step 5: Else if salary is between 15000 and 20000, print "Post: Manager".
Step 6: Else if salary is between 8000 and 15000, print "Post: Accountant".
Step 7: Else if salary is between 5000 and 8000, print "Post: Office boy".
Step 8: Else, print "Invalid salary range".
Step 9: Print "coded by pranjal khatri 929".
Step 10: End.

Problem 12



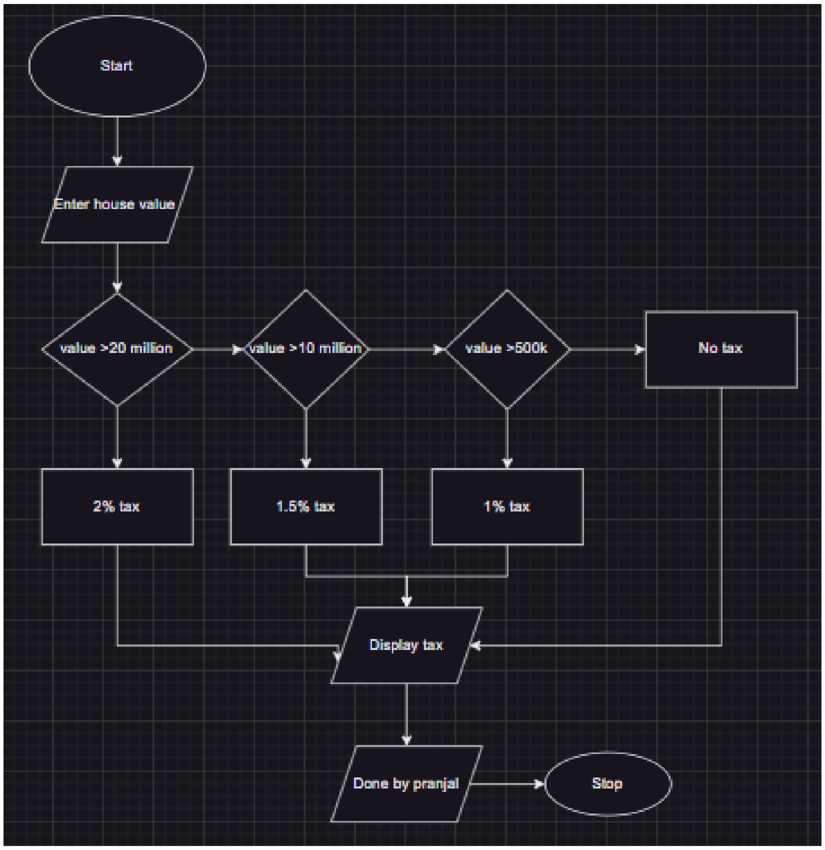
```c
/*
tax paying as per house value.
*/

#include <stdio.h>

int main() {
    int houseCount = 5;
    float houseValue, tax;

    for (int i = 1; i <= houseCount; i++) {
        printf("Enter the value of house %d: ", i);
        scanf("%f", &houseValue);


        if (houseValue > 2000000) {
            tax = houseValue * 0.02;  // 2% tax for houses over 2 million
        }
        else if (houseValue > 1000000) {
            tax = houseValue * 0.015; // 1.5% tax for houses over 1 million
        }
        else if (houseValue > 500000) {
            tax = houseValue * 0.01;  // 1% tax for houses over 500k
        }
        else {
            tax = 0;  // No tax for houses 500k or less
        }

        printf("The tax for house %d with value %.2f is: %.2f\n", i, houseValue, tax);
    }

    printf("Done by Pranjal khatri");
    return 0;
}
```

Output



```
Enter the value of house 1: 100000
The tax for house 1 with value 100000.00 is: 0.00
Enter the value of house 2: 10000000000
The tax for house 2 with value 10000000000.00 is: 200000000.00
Enter the value of house 3: 2323123
The tax for house 3 with value 2323123.00 is: 46462.46
Enter the value of house 4: 233235
The tax for house 4 with value 233235.00 is: 0.00
Enter the value of house 5: 3456
The tax for house 5 with value 3456.00 is: 0.00
Done by Pranjal khatri

...Program finished with exit code 0
Press ENTER to exit console.
```

Flowchart



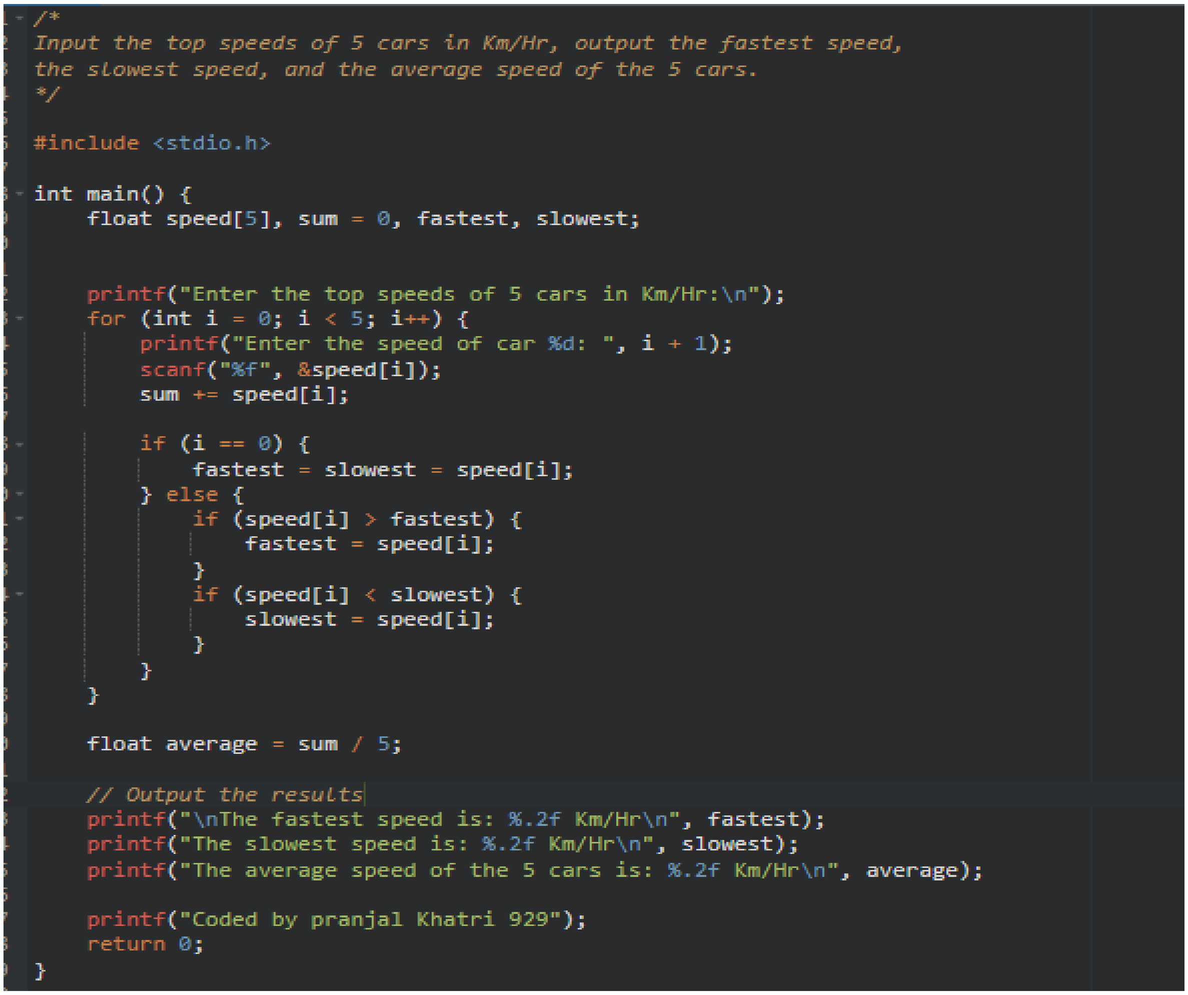
Algorithm

Step 1: Start.
Step 2: Set houseCount = 5 for the number of houses.
Step 3: For each house (from 1 to 5):

- Input the value of the house houseValue.
- If houseValue is greater than 2 million, calculate tax as 2% of the house value.
- Else if houseValue is greater than 1 million, calculate tax as 1.5% of the house value.
- Else if houseValue is greater than 500k, calculate tax as 1% of the house value.
- Else, set tax as 0.

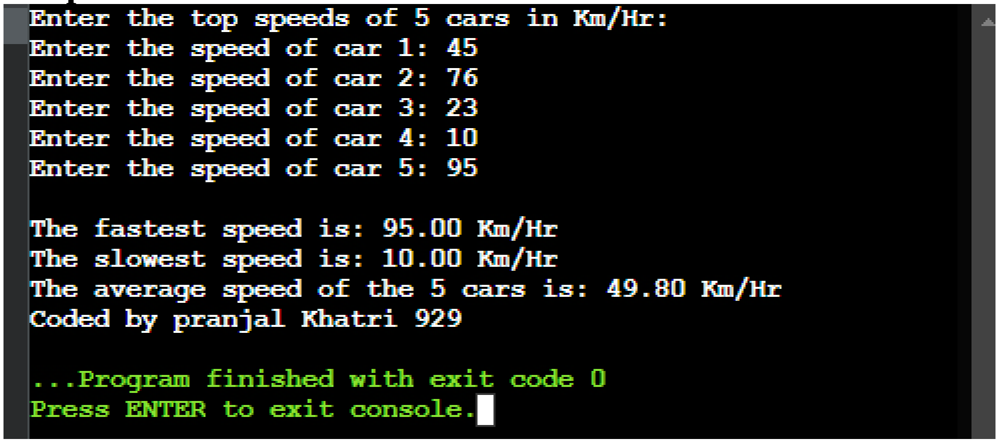
Step 4: Print the tax for the house.
Step 5: After processing all houses, print "Done by Pranjal Khatri".
Step 6: End.

Problem 13



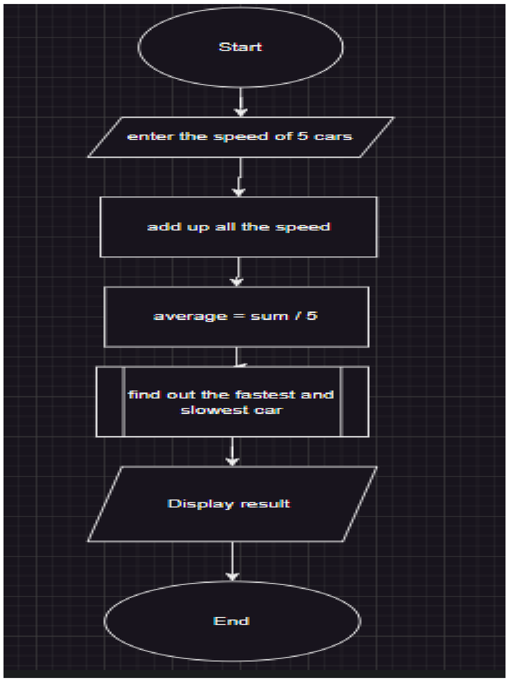
```c
/*
Input the top speeds of 5 cars in Km/Hr, output the fastest speed,
the slowest speed, and the average speed of the 5 cars.
*/

#include <stdio.h>

int main() {
    float speed[5], sum = 0, fastest, slowest;


    printf("Enter the top speeds of 5 cars in Km/Hr:\n");
    for (int i = 0; i < 5; i++) {
        printf("Enter the speed of car %d: ", i + 1);
        scanf("%f", &speed[i]);
        sum += speed[i];

        if (i == 0) {
            fastest = slowest = speed[i];
        } else {
            if (speed[i] > fastest) {
                fastest = speed[i];
            }
            if (speed[i] < slowest) {
                slowest = speed[i];
            }
        }
    }

    float average = sum / 5;

    // Output the results
    printf("\nThe fastest speed is: %.2f Km/Hr\n", fastest);
    printf("The slowest speed is: %.2f Km/Hr\n", slowest);
    printf("The average speed of the 5 cars is: %.2f Km/Hr\n", average);

    printf("Coded by pranjal Khatri 929");
    return 0;
}
```

Output



```
Enter the top speeds of 5 cars in Km/Hr:
Enter the speed of car 1: 45
Enter the speed of car 2: 76
Enter the speed of car 3: 23
Enter the speed of car 4: 10
Enter the speed of car 5: 95

The fastest speed is: 95.00 Km/Hr
The slowest speed is: 10.00 Km/Hr
The average speed of the 5 cars is: 49.80 Km/Hr
Coded by pranjal Khatri 929

...Program finished with exit code 0
Press ENTER to exit console.
```

Flowchart


Start
enter the speed of 5 cars
add up all the speed
average = sum / 5
find out the fastest and slowest car
Display result
End

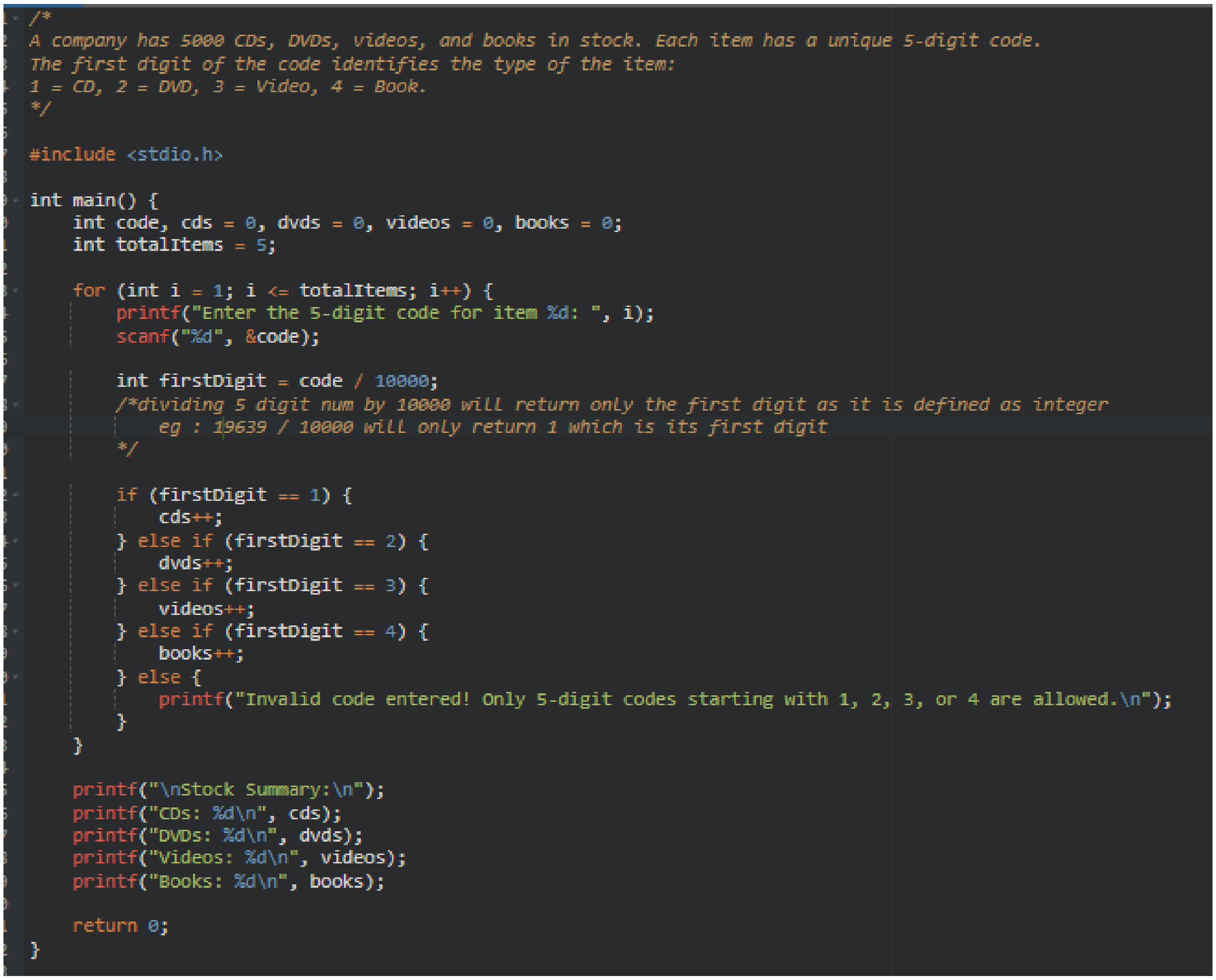
Algorithm

Step 1: Start.
Step 2: Declare an array `speed[5]`, and variables `sum`, `fastest`, and `slowest`.
Step 3: For each of the 5 cars:

- Input the speed of the car and store it in `speed[i]`.
- Add the speed to `sum`.
- If it's the first car, set `fastest` and `slowest` to `speed[i]`.
- Else, compare the speed with `fastest` and `slowest` and update accordingly.

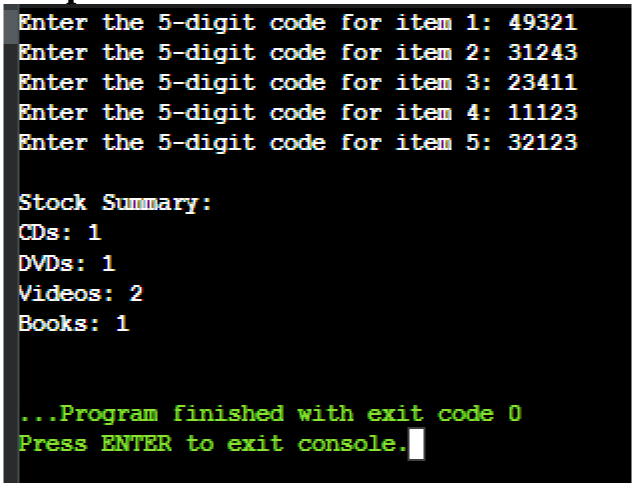
Step 4: Calculate the average speed by dividing `sum` by 5.
Step 5: Print the fastest speed, slowest speed, and average speed.
Step 6: Print "Coded by pranjal khatri 929".
Step 7: End.

Problem 14



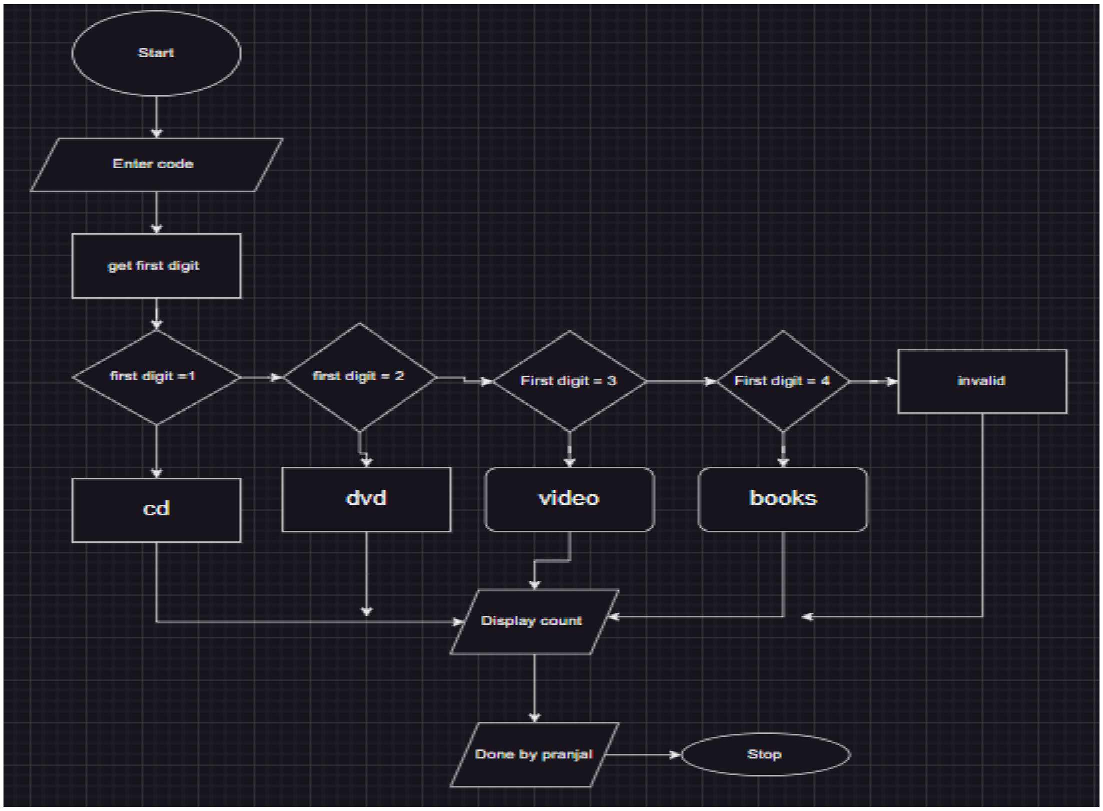
```c
/*
A company has 5000 CDs, DVDs, videos, and books in stock. Each item has a unique 5-digit code.
The first digit of the code identifies the type of the item:
1 = CD, 2 = DVD, 3 = Video, 4 = Book.
*/

#include <stdio.h>

int main() {
    int code, cds = 0, dvds = 0, videos = 0, books = 0;
    int totalItems = 5;

    for (int i = 1; i <= totalItems; i++) {
        printf("Enter the 5-digit code for item %d: ", i);
        scanf("%d", &code);

        int firstDigit = code / 10000;
        /*dividing 5 digit num by 10000 will return only the first digit as it is defined as integer
            eg : 19639 / 10000 will only return 1 which is its first digit
        */

        if (firstDigit == 1) {
            cds++;
        } else if (firstDigit == 2) {
            dvds++;
        } else if (firstDigit == 3) {
            videos++;
        } else if (firstDigit == 4) {
            books++;
        } else {
            printf("Invalid code entered! Only 5-digit codes starting with 1, 2, 3, or 4 are allowed.\n");
        }
    }

    printf("\nStock Summary:\n");
    printf("CDs: %d\n", cds);
    printf("DVDs: %d\n", dvds);
    printf("Videos: %d\n", videos);
    printf("Books: %d\n", books);

    return 0;
}
```

Output



```
Enter the 5-digit code for item 1: 49321
Enter the 5-digit code for item 2: 31243
Enter the 5-digit code for item 3: 23411
Enter the 5-digit code for item 4: 11123
Enter the 5-digit code for item 5: 32123

Stock Summary:
CDs: 1
DVDs: 1
Videos: 2
Books: 1


...Program finished with exit code 0
Press ENTER to exit console.
```

Flowchart



Algorithm

Step 1: Start.
Step 2: Initialize variables `cds`, `dvds`, `videos`, `books`, and `totalItems = 5`.
Step 3: For each of the 5 items (from 1 to 5):

- Input the 5-digit `code`.
- Extract the first digit of the code using `firstDigit = code / 10000`.
- If `firstDigit == 1`, increment `cds`.
- If `firstDigit == 2`, increment `dvds`.
- If `firstDigit == 3`, increment `videos`.
- If `firstDigit == 4`, increment `books`.
- If the first digit is not between 1 and 4, print an error message.

Step 4: After processing all items, print the stock summary
Step 5: End.

Conclusion

During this computer lab session, I effectively finished a set of programming tasks in the C language, significantly improving my grasp of essential programming concepts like conditional statements, loops, arrays, and functions. These tasks addressed a variety of real-life issues, such as testing divisibility by numbers, identifying triangle types, computing taxes based on property values, and classifying items with distinct identification codes. Addressing these challenges enabled me to utilize theoretical knowledge in real-world scenarios, honing my problem-solving skills and enhancing my coding expertise. I discovered how to apply several algorithms, manage various kinds of data, and generate the expected outputs effectively.

Moreover, this lab session offered me practical experience in troubleshooting and verifying my programs, guaranteeing their accuracy and dependability. I discovered how to detect mistakes in my coding, resolve problems, and enhance my applications for improved efficiency. Creating algorithms for every problem and illustrating them with flowcharts enabled me to more effectively visualize the logical processes required to tackle each task. This method not only streamlined the implementation process but also enhanced my skills in critical thinking and deconstructing complex issues into smaller, more manageable components. In general, this lab experience has greatly equipped me for more complex programming tasks and has established a strong basis for addressing practical software development projects ahead.