Program-1:

Implement a C program to find the distance between two points/coordinates

Formula:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Program code:

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

int main() {
    float x1, y1, x2, y2, distance;

    printf("Enter the first point coordinates  = ");
    scanf("%f %f", &x1, &y1);

    printf("Enter the second point coordinates  = ");
    scanf("%f %f", &x2, &y2);

   float  X = pow(x2 - x1, 2);
   float Y =  pow(y2 - y1, 2);
   distance = sqrt( X + Y );

    printf("The distance between two points is: %.2f\n", distance);

    return 0;
}
```

Sample output:

```
Enter the first point coordinates  = 10 20
Enter the second point coordinates  = 20 10
The distance between two points is: 14.14
```

Program-2:

Illustrate conditional branching statements to fnd the smallest of tree numbers

Program code:

```
#include <stdio.h>

int main() {
   int a, b, c, smallest;

   printf("Enter three numbers: ");
   scanf("%d %d %d", &a, &b, &c);

      if (a < b) {
      if (a < c) {
         smallest = a;
      } else {
         smallest = c;
      }
   } else {
      if (b < c) {
         smallest = b;
      } else {
         smallest = c;
      }
   }

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

   return 0;
}
```

Sample output:

```
Enter three numbers: 15 5 41
The smallest number is: 5
```

Program-3:

Write a C program to find all possible roots of a quadratic equation

quadratic equation: $ax^2+bx+c=0$

The program calculates the roots based on the discriminant, $D=b^2-4ac$

1. If D>0, the equation has two distinct real roots.
2. If D=0, the equation has one real root (a repeated root).
3. If D<0, the equation has two complex roots.

Program code:

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

int main() {
   float a, b, c, discriminant, root1, root2, realPart, imaginaryPart;

   printf("Enter coefficients a, b and c: ");
   scanf("%f %f %f", &a, &b, &c);

   discriminant = b * b - 4 * a * c;

   if (discriminant > 0) {
      root1 = (-b + sqrt(discriminant)) / (2 * a);
      root2 = (-b - sqrt(discriminant)) / (2 * a);
      printf("Roots are real and distinct: %.2f and %.2f\n", root1, root2);
   }
   else if (discriminant == 0) {
      root1 = -b / (2 * a);
      printf("Root is real and repeated: %.2f\n", root1);
   }
   else {
      realPart = -b / (2 * a);
      imaginaryPart = sqrt(-discriminant) / (2 * a);
      printf("Roots are complex: %.2f + %.2fi and %.2f - %.2fi\n", realPart,
            imaginaryPart, realPart, imaginaryPart);
   }

   return 0;
}
```

Sample output:

Enter coefficients a, b and c: 4 5 1
Roots are real and distinct: -0.25 and -1.00

Enter coefficients a, b and c: 2 4 2
Root is real and repeated: -1.00

Enter coefficients a, b and c: 2 3 2
Roots are complex: -0.75 + 0.66i and -0.75 - 0.66i