PROBLEM 1:

AIM : Write a C Program for Quadratic Equation Roots.

ALGORITHM :

- Start.
- Read a, b, c values.
- Compute d = b2 4ac.
- if d > 0 then. r1 = b+ sqrt (d)/(2*a) r2 = b sqrt(d)/(2*a)
- Otherwise if d = 0 then. compute r1 = -b/2a, r2=-b/2a. print r1,r2 values.
- Otherwise if d < 0 then print roots are imaginary.
- Stop**.**

PROGRAM CODE:

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

// equation ax*2 + bx + x
void findRoots(int a, int b, int c)
{
   // If a is 0, then equation is linear
     if (a == 0) {
       printf("Invalid input\nYou have entered a linear equation");
       return;
   }
  int d = b * b - 4 * a * c;
   double sqrt_val = sqrt(abs(d));

   if (d > 0) {
       printf("Roots are real and different\n");
       printf("%f\n%f", (double)(-b + sqrt_val) / (2 * a),  (double)(-b - sqrt_val) / (2 * a));
```

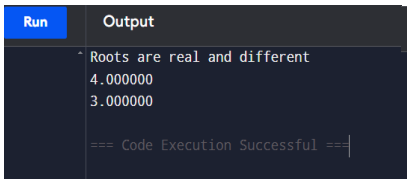
```
    }
    else if (d == 0) {
        printf("Roots are real and same\n");
        printf("%f", -(double)b / (2 * a));
    }
 // d < 0
   else   {
        printf("Roots are complex\n");
        printf("%f + i%f\n%f - i%f", -(double)b / (2 * a), sqrt_val / (2 * a), -(double)b / (2 * a), sqrt_val / (2 * a));  } }
// Driver code
int main()
{
    int a , b , c ;
printf(“Enter the coefficients of the quadratic equation (a,b,c): “);
scanf(“%d %d %d”,&a,&b,&c);
// Function call
    findRoots(a, b, c);
    return 0;
}
```

REFUTE TEST CASE :

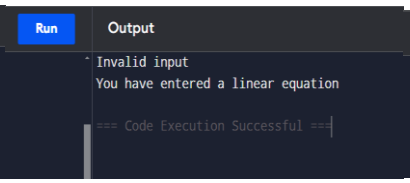
- The entered coefficient of a is 0 : The compiler will give a “Invalid input” message.
- The Roots are not real : The compiler will give a message “Complex roots”.

EXPECTED OUTPUTS :

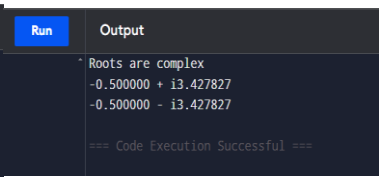
TEST RUN 1:



TEST RUN 2 :



TEST RUN 3 :



RESULT : The program was run and executed successfully.

PROBLEM : 2

AIM : Write a C program to find the derivative of a polynomial.

ALGORITHM :

- For each term in the polynomial, multiply the term's coefficient by its power.
- Decrease the power by 1.
- Repeat for all terms in the polynomial.

PROGRAM CODE :

```
#include <stdio.h>
int main() {
   int deg,i,coeff;
   printf("Enter the degree of polynomial: ");
   scanf("%d",&deg);
   if (scanf("%d")) {
   int array[deg],arr[deg];
      for (i=0;i<=deg;i++) {
         printf("\nEnter the coefficient of x^%d: ",deg-i);
         scanf("%d",&coeff);
         array[i]=coeff; }
     for(i=0;i<=deg;i++) {
         arr[i]=((deg-i)*array[i]); }

      printf("Given equation : ");
      for(i=0;i<=deg;i++) {
         if(i==deg) {
            printf(" (%d) ",array[i]); }
         else {
            printf(" (%dx^%d) +",array[i],deg-i); } }
```

```
    printf("\nDifferentiated equation : ");
    for(i=0;i<deg;i++) {
      if(i==(deg-1)) {
        printf(" (%d) ",arr[i]); }
      else {
        printf(" (%dx^%d) +",arr[i],deg-i-1); } } }
    else {
      printf("Invalid Input");
    }
  return 0; }
```

REFUTE TEST CASE :

- Polynomial degree is entered zero : The compiler would just take input the end the program at no further message, thus a if statement is used to show a message of invalid input with appropriate explanation.
- Polynomial degree entered is negative : Invalid input will be displayed.
- Polynomial degree entered is not an integer : Invalid Input will be displayed.

TEST RUN 1:

```
Run
Output
Enter the degree of polynomial: 4.5
Invalid input

=== Code Execution Successful ===
```

TEST RUN 2 :

```
Run
Output
Clear
Enter the degree of polynomial: 0
Invalid Input
Degree of the polynomial 0 gives the equation to be 1 and on further
    differentiation the derivative would be 0 as well

=== Code Execution Successful ===
```

TEST RUN 3:

```
Run
Output
Enter the degree of polynomial: 3

Enter the coefficient of x^3: 3

Enter the coefficient of x^2: 4

Enter the coefficient of x^1: 5

Enter the coefficient of x^0: 6
Given equation :  (3x^3) + (4x^2) + (5x^1) + (6)
Differentiated equation :  (9x^2) + (8x^1) + (5)

=== Code Execution Successful ===
```

RESULT : The program was run and executed successfully.