31/08/2020 **Mohd. Aftab Alam**
02013302717

EXPERIMENT 2

Aim :

(a) To determine the nature of roots of a Quadratic Equation, its input is triple of positive integers (say a, b, c) and values may be from interval [1, 100]. The output may have one of the following :

Real & Distinct Roots, Imaginary Roots or Real & Equal Roots.

Design the Boundary Value Test Cases.

Algorithm :

- Take 3 inputs from the user for the quadratic equation.
- Check whether they lie in the given interval.
- If the condition is false, stop the program and exit.
- Else if condition is true, check the nature of the roots.
- If the Discriminant is greater than 0, Real & Distinct Roots.
- If the Discriminant is less than 0, Imaginary Roots.
- If the Discriminant is equal to 0, Real & Equal Roots.
- According to the formula ***4n+1,*** there will be 13 test cases, where n is number of inputs.

Code :

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

using namespace std;

int bva(int, int, int);

int main()
{
    int min, max;
    int x, y, z;
    cout << "Enter Range : ";
    cin >> min >> max;
    if (min < 0 || max > 100)
    {
        cout << "Invalid Range";
        return 0;
    }
    int nominal = (min + max) / 2;
    int values[] = {min, min + 1, nominal, max - 1, max};
    cout << "a\tb\tc\tOutput\t\t\tRoots" << endl;
    for (int i = 0; i < 5; i++)
    {
        bva(values[i], nominal, nominal);
    }
```

```
    for (int i = 0; i < 5; i++)
    {
        if (values[i] != nominal)
            bva(nominal, values[i], nominal);
    }
    for (int i = 0; i < 5; i++)
    {
        if (values[i] != nominal)
            bva(nominal, nominal, values[i]);
    }

    cout << "Enter the Coefficients (a, b, c) : ";
    cin >> x >> y >> z;
    cout << "a\tb\tc\tOutput\t\t\tRoots" << endl;
    bva(x, y, z);
    return 0;
}

int bva(int a, int b, int c)
{
    if (a == 0)
    {
        cout << "Not a Quadratic Equation\n";
        return 0;
    }
    int d = b * b - 4 * a * c;
    double sqrt_val = sqrt(abs(d));
    cout << a << "\t" << b << "\t" << c << "\t";
    if (d < 0)
    {
        cout << "Imaginary Roots\t\t";
        cout << -(double)b / (2 * a) << "+i" << sqrt_val << ", ";
        cout << -(double)b / (2 * a) << "-i" << sqrt_val << endl;
    }
    else if (d == 0)
    {
        cout << "Real and Equal\t\t";
        cout << -(double)b / (2 * a) << endl;
    }
    else
    {
        cout << "Real and Distinct\t";
        cout << (double)(-b + sqrt_val) / (2 * a);
        cout << ", " << (double)(-b - sqrt_val) / (2 * a) << endl;
    }
    return 0;
}
```

31/08/2020 **Mohd. Aftab Alam**
02013302717

Boundary Value Analysis :

Range : R [1, 100]

Domain : Minimum = 1

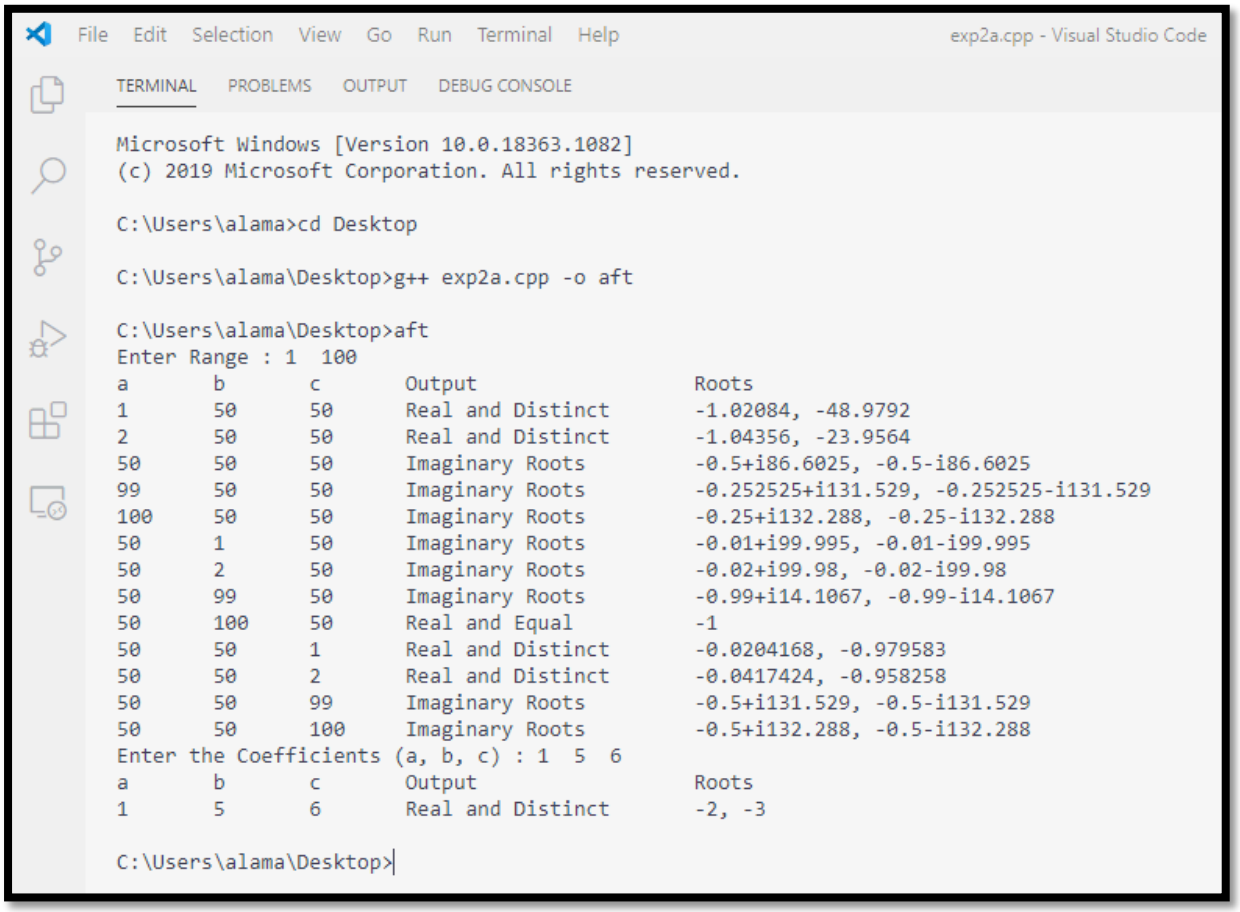
Above Minimum = 2

Nominal = 50

Below Maximum = 99

Maximum = 100

Output :



Aim :

(b) Consider a program for determining the Previous Date. Its input is triple of Day, Month and Year with values in the range :

$$1 \leq \text{Day} \leq 31$$

$$1 \leq \text{Month} \leq 12$$

$$1900 \leq \text{Year} \leq 2025$$

Possible outputs would be Previous Date or Invalid Date. Design the Boundary Value Test Cases.

Algorithm :

- Take 3 inputs from the user for Day, Month and Year.
- Check whether they lie in the given intervals.
- If the condition is false, stop the program and exit.
- If the condition is true, calculate the date according to the given values.
- Subtract 1 day from it to get the Previous Date.
- According to the formula ***4n+1,*** there will be 13 test cases, where n is number of inputs.

Code :

```
#include <iostream>
using namespace std;
void bva(int, int, int);
int main()
{   int amin, amax, bmin, bmax, cmin, cmax;
    int x, y, z;
    cout << "Enter Range for Day : ";
    cin >> amin >> amax;
    cout << "Enter Range for Month : ";
    cin >> bmin >> bmax;
    cout << "Enter Range for Year : ";
    cin >> cmin >> cmax;
    if (amin < 1 || amax > 31)
    {
        cout << "Invalid Day Range";
        return 0;
    }
    if (bmin < 1 || bmax > 12)
    {
        cout << "Invalid Month Range";
        return 0;
    }
    if (cmin < 1900 || amax > 2025)
    {
        cout << "Invalid Year Range";
        return 0;
    }
```

```
    int anominal = (amin + amax) / 2;
    int avalues[] = {amin, amin + 1, anominal, amax - 1, amax};
    int bnominal = (bmin + bmax) / 2;
    int bvalues[] = {bmin, bmin + 1, bnominal, bmax - 1, bmax};
    int cnominal = (cmin + cmax) / 2;
    int cvalues[] = {cmin, cmin + 1, cnominal, cmax - 1, cmax};
    cout << "Day\tMonth\tYear\tExpected Output" << endl;
    for (int i = 0; i < 5; i++)
        bva(avalues[i], bnominal, cnominal);
    for (int i = 0; i < 5; i++)
        bva(anominal, bvalues[i], cnominal);
    for (int i = 0; i < 5; i++)
        bva(anominal, bnominal, cvalues[i]);
    cout << "Enter the Day, Month and Year : ";
    cin >> x >> y >> z;
    cout << "Day\tMonth\tYear\tExpected Output" << endl;
    bva(x, y, z);
    return 0;
}

void bva(int a, int b, int c)
{
    cout << a << "\t" << b << "\t" << c << "\t";
    if (a != 1)
    {
        if ((b == 2 || b == 4 || b == 6 || b == 9 || b == 11) & (a == 31))
            cout << "Invalid Date" << endl;
        else if ((b == 2) & (a == 30))
            cout << "Invalid Date" << endl;
        else if ((b == 2) & (a == 29) & (c % 4 != 0))
            cout << "Invalid Date" << endl;
        else
            cout << a - 1 << "-" << b << "-" << c << endl;
    }
    else
    {
        if (b == 3)
        {
            if (c % 4 == 0)
                a = 29;
            else
                a = 28;
        }
        cout << a << "-" << b - 1 << "-" << c << endl;
    }
}
```

Boundary Value Analysis :

Range : R [1, 31] [1, 12] [1900, 2025]

Domain : Minimum = 1, 1, 1900

Above Minimum = 2, 2, 1901

Nominal = 16, 6, 1962

Below Maximum = 30, 11, 2024

Maximum = 31, 12, 2025

Output :