Name: Shu Xiangyu
Student Number: 222021321062016

Question_1:

1. Get the last four digits (substring to the number)of the student number and store it as *Seed* .
2. Use the *Seed* to generate random numbers within [1, 100] for the elements of the array.
3. The Array's length depends on the requisition.(last two digits mod 20)
4. Use the **Bubble Sort** to achieve descending order or ascending order for method1.
5. Using the **Quick Sort** (the recursive method)for method2.

Method 1 (Bubble Sort)

Source Code:

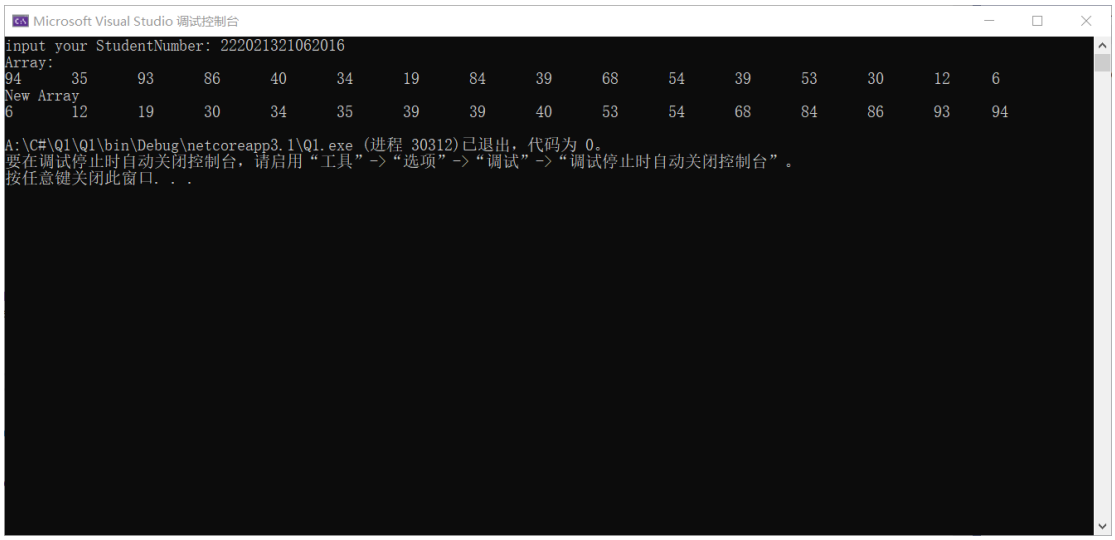
```
using System;
public class Method
{
    public int[] GetArray(ref string StudentNumber)
    {
       int seed = Convert.ToInt32(StudentNumber.Substring(StudentNumber.Length-4,4));
        var rd = new Random(seed);//Seed



       int len = Convert.ToInt32(StudentNumber.Substring(StudentNumber.Length - 2, 2));
       if (len%20<=5)
        {
            len += 10;
        }
        int[] Array = new int[len];
        int i;
        for(i=0 ;i<Array.Length ;i++ )
        {
            Array[i] = rd.Next(0, 101);
        }//generate the array
        Console.WriteLine("Array: ");
        for(i=0;i<Array.Length ;i++)
        {
            Console.Write("{0}\t", Array[i]);
```

```
    }
    Console.WriteLine();

    return (Array);
}


 public void descending(int[] Array)//descending
{
    int i, j;
    for(i=0;i<Array.Length-1 ; i++)
    {
        for(j=0;j<Array.Length-1-i ; j++)
        {
            if (Array[j] < Array[j +1])
            {
                int tamp;
                tamp = Array[j + 1];
                Array[j + 1] = Array[j];
                Array[j] = tamp;
            }
        }
    }

    Console.WriteLine("New Array");
    for(i=0;i<Array.Length ;i++)
    {
        Console.Write("{0}\t", Array[i]);
    }
    Console.WriteLine();


}


public  void ascending(int[] Array)//ascending
{
    int i, j;
    for (i = 0; i < Array.Length - 1; i++)
    {
        for (j = 0; j < Array.Length - 1 - i; j++)
        {
            if (Array[j] > Array[j + 1])
            {
```

```
                    int tamp;
                    tamp = Array[j+1 ];
                    Array[j + 1] = Array[j];
                    Array[j] = tamp;
                }
            }
        }

        Console.WriteLine("New Array");
        for (i = 0; i < Array.Length; i++)
        {
            Console.Write("{0}\t", Array[i]);
        }
        Console.WriteLine();
    }

    public void DisPlay()
    {
        Console.Write("input your StudentNumber: ");
        string StudentNumber = Console.ReadLine();









        int i = Convert.ToInt32(StudentNumber.Substring(0, 1));
        if (i % 2 == 1)
        {
            descending(GetArray(ref StudentNumber));
        }
        else
        {
           ascending(GetArray(ref StudentNumber));
        }
    }

}
public class Program
{
    public static void Main()
    {
        Method m1 = new Method();
        m1.DisPlay();
```

```
    }
}
```

Result:



AS we have sort the array according to the ascending order, in method2 we should sort the array according the descending order.

Method 2 (Quick Sort)

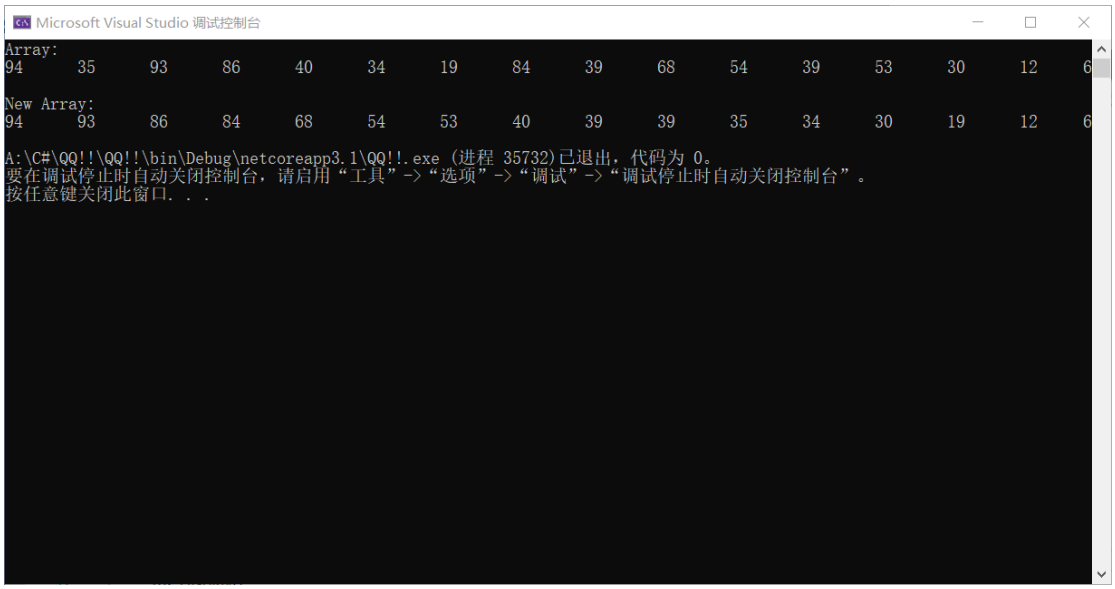
Source Code:

```
using System;

public class Method2
{

    //QuickSort method.(descending)
    public int quicksort(int[] array, int left, int right)
    {
        int tamp = array[left];

        while (left < right)
        {
            while (left < right && array[right] <= tamp)
            {
                right--;
            }
```

```
            array[left] = array[right];

            while (left < right && array[left] >= tamp)
            {
                left++;
            }

            array[right] = array[left];
        }
        array[left] = tamp;
        return right;
    }
     public void QuickSort(int[] array, int left , int right)
    {
        if(left >=right)
        {
            return ;
        }
       int  param = quicksort(array,left,right);
        QuickSort(array, left, param - 1);
        QuickSort(array, param + 1, right);



    }


    public void Display()
    {
        Console.WriteLine("Enter your student Number: ");
        string StudentNumber = Console.ReadLine();
        int seed = Convert.ToInt32(StudentNumber.Substring(StudentNumber.Length - 4,
4));
        var rd = new Random(seed);//Seed



        int len = Convert.ToInt32(StudentNumber.Substring(StudentNumber.Length - 2,
2));
        if (len % 20 <= 5)
        {
            len += 10;
        }
        int[] Array = new int[len];
        int i;
```

```
    for (i = 0; i < Array.Length; i++)
    {
        Array[i] = rd.Next(0, 101);
    }//generate the array


    Console.WriteLine("Array: ");
    for (i = 0; i < Array.Length; i++)
    {
        Console.Write("{0}\t", Array[i]);
    }
    Console.WriteLine();//Display the original array


    QuickSort(Array, 0, Array.Length - 1); //Quick Sort

    Console.WriteLine("New Array:");
       for( i=0; i<Array.Length;i++)
    {
        Console.Write("{0}\t",Array[i]);
    }


}



static void Main(string[] args)

    {
    Method2 m2 = new Method2();
    m2.Display();




}
}
```

Result:



Question_2:

1. Get the last four digits (**substring** to the number)of the student number and store it as *Seed* .
2. Use the *Seed* to generate 3 random numbers, as *a* , *b* , *c* .
3. Use the Newton method, **fine the fixed point of g(x) = x – f(x) / df(x).**
4. Set the Precision and use **Iteration** to get the result

Source Code:

```
using System;

namespace Q_2
{
  public  class Func
    {

        public double func(ref int a ,ref int b,ref int c,ref double Varible)
        {
            double Value = Varible -( a * Varible * Varible + b * Varible +
c)/(2*a*Varible+b);

            return Value;
        }


        static void Main(string[] args)
```

```
        {
            Console.Write("Input your Student Number: ");
            string StudentNumber = Console.ReadLine();
            int Seed = Convert.ToInt32(StudentNumber.Substring(StudentNumber.Length -
4, 4));
            var rd = new Random(Seed);
            int[] Array = new int[3];
            int i;//generate the array




            Func f = new Func();
            double guess=1.0, next=1.0;
                int a, b, c;
            do
            {
                for (i = 0; i < Array.Length; i++)
                {
                    Array[i] = rd.Next(0, 100);
                }

                a = Array[0];
                b = Array[1];
                c = Array[2];

                if(f.func(ref a ,ref b, ref c,ref guess)==0)
                {

                    Console.WriteLine("result ={0}",guess);
                    break;
                }
                for (i = 0; i < 10; i++)
                {
                    guess = next;
                    next = f.func(ref a, ref b, ref c, ref guess);

                }

                Console.WriteLine();
            } while (Math.Abs(guess - next) > 0.0001);

            Console.WriteLine("a = {0}\tb={1}\tc={2};", a, b, c);
```

```
            Console.WriteLine("final result: {0}",next);
        }
    }
}
```

Result: