Kim, Jun

Satish Singhal

CSCS-1 #0105

Mar. 18, 2021

Assignment 3: IOAA Document

**Data Input**

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| A | int | Coefficient A in formulae |
| B | int | Coefficient B in formulae |
| C | int | Coefficient C in formulae |

**Data Output**

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| X1 | double | Result of formulae when A and B is not zero |
| X2 | double | Result of formulae when A and B is not zero |
| X3 | double | Result of formulae when A is zero |
| negDiscriminant | double | Result of formulae when discriminant is negative |

**Computational Aid and Other Variables**

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| discriminant | int | Check for Nature of Solutions |
| negDiscriminant | double | calculation value when discriminant is negative (sqrt can’t calculate negative value) |
| flagA | bool | Check for coefficient A value is not zero |
| flagB | bool | Check for coefficient B value is not zero |

**Global Constants**

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remark |
| endMessage | string | End message when finishing program |

**Analysis**

|  |  |  |
| --- | --- | --- |
| Math Formulae | C++ Formulae | Remark |
| (-B + (B2 – 4AC)1/2) / 2A | (-B + (B\*B + sqrt(B\*B - 4\*A\*C)) / (2.0\*A) | X1 |
| (-B - (B2 – 4AC)1/2) / 2A | (-B - (B\*B + sqrt(B\*B - 4\*A\*C)) / (2.0\*A) | X2 |
| - C / B | - C / B | X3 |
| B2 – 4AC | B\*B - 4\*A\*C | Discriminant |

flagA = (A != 0)

flagB = (B != 0)

discriminant = B\*B – 4\*A\*C

If (flagA) then

|  |  |  |
| --- | --- | --- |
| Boolean Condition and value | Algorithmic Task | Remark |
| flagA == true and discriminant > 0 | X1 = (-B + sqrt(discriminant)) / (2.0\*A)  X2 = (-B - sqrt(discriminant)) / (2.0\*A) |  |
| flagA == true and discriminant < 0 | X1 = -B / (2.0\*A)  X2 = -B / (2.0\*A)  negDiscriminant = sqrt(abs(discriminant)) / (2.0\*A) | flagB == true |
| X1 = 0  X2 = 0  negDiscriminant = sqrt(abs(discriminant)) / (2.0\*A) | flagB == false  X1, and X2 value are not used |
| flagA == true and discriminant == 0 | X1 = (-B + sqrt(discriminant)) / (2.0\*A)  X2 = 0 | X2 value is not used |

Else

|  |  |  |
| --- | --- | --- |
| Boolean Condition and value | Algorithmic Task | Remark |
| flagA == false and  flagB == true | X3 = - C / B | A value is used zero |
| flagA == false and  flagB == false | Error message | both A and B value are used zero |

End if

**Algorithm**

1. Add all the #include directives, declare global constants
2. declare A as int
3. declare B as int
4. declare C as int
5. declare X1 as double
6. declare X2 as double
7. declare X3 as double
8. declare flagA as bool
9. declare flagB as bool
10. declare discriminant as int
11. Print greeting message and explain formulae
12. prompt user for value A
13. get and store input into variable A
14. set boolean flag for A type as bool flagA = (A != 0)
15. If (flagA) then
    1. prompt user for value B
    2. get and store input into variable B
    3. set boolean flag for B type as bool flagB = (B != 0)
    4. prompt user for value C
    5. get and store input into variable C
    6. discriminant = B\*B – 4\*A\*C
    7. If (discriminant > 0) then
       1. X1 = (-B + sqrt(discriminant)) / (2.0\*A)
       2. X2 = (-B - sqrt(discriminant)) / (2.0\*A)
       3. Print, “Two Real Solutions”, X1, X2, EOL
       4. print end message
    8. Else if (discriminant < 0) then
       1. declare negDiscriminant as double
       2. negDiscriminant = sqrt(abs(discriminant)) / (2.0\*A)
       3. If (flagB) then
          1. X1 = -B / (2.0\*A)
          2. X2 = -B / (2.0\*A)
          3. Print,”Two Imaginary Solutions”,X1,”+(“,negDiscriminant,”)\*I”

,EOL,X2,”-(“,negDiscriminant,”)\*I”,EOL

* + - 1. print end message
    1. Else
       1. Print,”Two Imaginary Solutions”,”(“,negDiscriminant,”)”,EOL

,”-(“,negDiscriminant,”)”,EOL

* + - 1. print end message
  1. Else
     1. X1 = (-B + sqrt(discriminant)) / (2.0\*A)
     2. Print, “One Real Solution”, X1, EOL
     3. print end message

1. Else
   1. Print warning and ask calculation without value A
   2. prompt user for value B
   3. get and store input into variable B
   4. set boolean flag for B type as bool flagB = (B != 0)
   5. If (flagB) then
      1. prompt user for value C
      2. get and store input into variable C
      3. X3 = - C / static\_cast<double>(B)
      4. Print, “Solution”, X3,EOL
      5. print end message
   6. Else
      1. print error message
      2. print end message
2. End if