

# 2019년 컴퓨터 프로그래밍 3

- HW 02 -

이차 방정식 풀기  
(Solving Quadratic Equations)

제출일자	~ 2019.03.20
이름	김용준
학번	201400875
분반	03

# 1. 프로그램 설명서

2차방정식의 근을 구하는 프로그램으로, 사용자에게 2차항, 1차항, 상수항의 계수를 입력 받으면 그것의 근의 존재여부와 2차항이 제대로 입력이 되었는지를 판단한 후, 실근 존재여부의 문제가 없으면 그 2차방정식의 근을 출력해주는 프로그램이다.

## 2. 생각해볼 점

첫 번째. 2차방정식의 2차항의 계수가 제대로 입력이 되었는지 확인해 본다. >> 왜냐하면, 2차항이 0일 때는 2차방정식이 아니게 되므로 주의해야한다.

두 번째. 2차방정식의 판별식. 즉,  $b^2 - 4ac$ 가  $< 0$ 일 때는 실근이 존재하지 않으므로 오류가 있다고 판단하여 사용자에게 알려주어야 한다.

세 번째. 정상적으로 판별식이  $\geq 0$  일 때 실근 2개 또는 1개(중근)을 정상적으로 출력하여야 한다.

### 3. 실행 결과

```
<<< 2차방정식 풀이를 시작합니다. >>>
방정식풀이를 시작하려면 'y', 풀이를 종료하려면 다른 아무 키나 치시오 :
y
2차항의 계수를 입력하시오: 1
1차항의 계수를 입력하시오: -4
상수항의 계수를 입력하시오: 4
<1.000000> x * x + <-4.000000> x + <4.000000> = 0
주어진 방정식의 해 x1 = 2.000000 x2 = 2.000000
방정식풀이를 시작하려면 'y', 풀이를 종료하려면 다른 아무 키나 치시오 :
y
2차항의 계수를 입력하시오: 1
1차항의 계수를 입력하시오: -5
상수항의 계수를 입력하시오: 6
<1.000000> x * x + <-5.000000> x + <6.000000> = 0
주어진 방정식의 해 x1 = 3.000000 x2 = 2.000000
방정식풀이를 시작하려면 'y', 풀이를 종료하려면 다른 아무 키나 치시오 :
y
2차항의 계수를 입력하시오: 1
1차항의 계수를 입력하시오: -3
상수항의 계수를 입력하시오: 9
<1.000000> x * x + <-3.000000> x + <9.000000> = 0
판별식의 값이 0보다 작습니다방정식풀이를 시작하려면 'y', 풀이를 종료하려면 다른
아무 키나 치시오 :
y
2차항의 계수를 입력하시오: 0
1차항의 계수를 입력하시오: 1
상수항의 계수를 입력하시오: 1
<1.000000> x + <1.000000> = 0
2차항의 계수가 0입니다. 제대로 입력해 주세요.
방정식풀이를 시작하려면 'y', 풀이를 종료하려면 다른 아무 키나 치시오 :
```

1. 중근체크
2. 정상적인 실근 2개 체크
3. 판별식이 0보다 작을 때 체크
4. 상수항이 0일 때 체크

## 4. 소스코드 분석

```
1 #include <math.h>
2 #include "AppView.h"
3
4
5 void main()
6 {
7     QuadEquationProblem qeProblem; // 객체 qeProblem 생성
8     Boolean SolvingIsRequested; // 논리할수 쓸 함수 정의
9     QuadEquation equation; // 객체 equation 생성
10    Solution solution; // 객체 solution 생성
11
12
13
14
15    AppView_Out_StartSolvingQuadEquation(); // 시작 메시지 출력함수 호출
16    SolvingIsRequested = AppView_In_SolvingIsRequested(); // y를 눌러 시작을 할지 의견을 묻는 함수 호출
17    while (SolvingIsRequested) { // SolvingIsRequested 값이 true이면,
18        AppView_In_QuadEquation(&equation._c0, &equation._c1, &equation._c2); // 2차방정식을 풀기 위한 계수를 입력 받고 (3개)
19
20        AppView_Out_QuadEquation(equation._c0, equation._c1, equation._c2); // 입력받은 2차방정식을 화면에 보여준다.
21        qeProblem._equation = equation; // 객체에
22
23        if (QuadEquationProblem_SecondOrderTermCoefficientIsZero(qeProblem)) { // 2차항의 계수를 판단하는 함수
24            AppView_Out_Msg_SecondOrderTermCoefficientIsZero(); // 2차항의 계수가 0일때 오류라고 알려주는 메시지 출력
25        }
26        else {
27            if (QuadEquationProblem_DeterminantIsNegative(qeProblem)) { // 판별식의 값을 판단하는 함수
28                Float determinant = QuadEquationProblem_Determinant(qeProblem); // 판별식의 값의 오류 여부를 저장
29                AppView_Out_Msg_DeterminantIsNegative(determinant); // 판별식의 값이 음수이면 오류라고 알려주는 메시지 출력
30            }
31            else {
32                solution = QuadEquationProblem_Solve(qeProblem); // solution이라는 객체에 2차방정식 풀이의 답을 저장
33                AppView_Out_Solution(solution._root1, solution._root2); // 2차방정식의 근 2개 출력
34            }
35        }
36
37
38
39        SolvingIsRequested = AppView_In_SolvingIsRequested(); // 2차방정식 풀이를 더 진행 할것지를 묻는 함수 호출
40
41        AppView_Out_Msg_EndSolvingQuadEquation(); // 종료 메시지 출력함수 호출
```