

# CS270 Recitation 1

## “C Programming Exercise”

### Goals

To modify, compile, and run a C program that solves quadratic equations.

### The Assignment

Start by making a directory called cs270 in your home directory

```
%> mkdir cs270
```

```
%> cd cs270
```

Make a subdirectory called R1 (inside cs270) for the recitation assignment, all files should reside in this subdirectory.

```
%> mkdir R1
```

```
%> cd R1
```

We have provided the framework of the C program to get you started. Open up `gedit` (or text editor of your preference), copy the code shown below, and save it into a file called `r1.c` in your R1 subdirectory.

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

// Function declaration
float quadratic(float a, float b, float c, bool returnFirstRoot);

// Function: quadratic
// Description: Solves a quadratic equation for one of its two roots.
// Parameters: a, b, and c represent the coefficients of a quadratic equation
// written in its standard form, i.e.,  $ax^2 + bx + c = 0$ .
// If returnFirstRoot is true, this function returns the first of the
// two roots. If false, it returns the other.
// Return: Returns one of the (possibly) two roots of the equation solution.
// Error: Return 0.0 if division by zero would occur.
float quadratic(float a, float b, float c, bool returnFirstRoot){

    // Avoid division by zero
    if (a == 0) return 0.0;

    float result = 0.0;
    // <Implement your quadratic equation solver here>
    return result;
}

// Program entry point
int main(){
    float a, b, c, root1, root2;
    printf("Quadratic Program\n");
    printf("Enter a: ");
    scanf("%f", &a);
    printf("Enter b: ");
    scanf("%f", &b);
    printf("Enter c: ");
    scanf("%f", &c);
    root1 = quadratic(a, b, c, true);
    root2 = quadratic(a, b, c, false);
    printf("Roots are %3.2f and %3.2f\n", root1, root2);
}
```

Compile the program into an executable called r1, as shown below.

```
%> gcc -g -std=c99 -Wall -lm r1.c -o r1
```

To run the compiled program, type the following command:

```
%> ./r1
```

Verify that the program always returns zero for the root, since the quadratic function is not implemented, then edit the program (using `gedit`) and implement the quadratic equation solver in `r1.c`. Recompile and run the program with the following test sets:

```
a = 1, b = 0, c = -25, should return roots are 5.00 and -5.00
```

```
a = 1, b = 1, c = -6, should return roots are 2.00 and -3.00
```

```
a = 1, b = -5, c = 4, should return roots are 4.00 and 1.00
```