

```

from math import *

# (4 - 4 + 4 - 4) is 0
print("In python, zero is: ", (4 - 4 + 4 - 4))

# (4 / 4) - (4 - 4) is 1
print("In python, one is: ", (4 / 4) - (4 - 4))

# (4 - 4) + (4 - sqrt(4)) is 2
print("In python, two is: ", (4 - 4) + (4 - sqrt(4)))

# (sqrt(4) + sqrt(4) - 4/4) is 3
print("In python, three is: ", (sqrt(4) + sqrt(4) - 4/4))

# 4 - 4 + 4 + sqrt(4) is 6
print("In python, six is: ", (4 - 4 + 4 + sqrt(4)))

# (sqrt(4) + sqrt(4) + sqrt(4) + sqrt(4)) is 8
print("In python, eight is: ", (sqrt(4) + sqrt(4) + sqrt(4) + sqrt(4)))

# (4! - sqrt(4) + sqrt(4) - 4)
print("In python, twenty is: ", (factorial(4) - sqrt(4) + sqrt(4) - 4))

```

Python version example 1. This seemed simple enough.

```

In [2]: run hw4pr3ex1.py
In python, zero is: 0
In python, one is: 1.0
In python, two is: 2.0
In python, three is: 3.0
In python, six is: 6.0
In python, eight is: 8.0
In python, twenty is: 20.0

```

My python outputs for example 1

```

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

int main() {
    int a, b, c, d, e, f, g;

    a = (sqrt(4) + sqrt(4) + 4 - 4); // four
    b = 4*4/4*4; //sixteen
    c = sqrt(4) + sqrt(4) + sqrt(4) + 4; //ten
    d = 4 * 4 - 4*1; //twelve
    e = 4/4 + sqrt(4) + sqrt(4); // five
    f = 4 * sqrt(4) + 4/4; //nine
    g = 4/4 + 4 + sqrt(4); // seven

    printf("In C, four is: %d\n",a);
    printf("In C, five is: %d\n",e);
    printf("In C, seven is: %d\n",g);
    printf("In C, nine is: %d\n",f);
    printf("In C, ten is: %d\n",c);
    printf("In C, twelve is: %d\n",d);
    printf("In C, sixteen is: %d\n",b);

    return 0;
}

```

```

PS C:\Users\pamor\Desktop\hw week 4> gcc .\hw4pr3ex1.c
PS C:\Users\pamor\Desktop\hw week 4> ./a.exe
In C, four is: 4
In C, five is: 5
In C, seven is: 7
In C, nine is: 9
In C, ten is: 10
In C, twelve is: 12
In C, sixteen is: 16
PS C:\Users\pamor\Desktop\hw week 4> 

```

Here are my outputs for the C version of example 1. Given my background, this wasn't bad at all. It brought back a lot of memories due to the similarities with C++ at this level.

```

from math import *

def Discriminant(a, b, c):
    d = b**2 - 4*a*c
    return d

def QuadSolver(a, b, c):
    xPlus = (-b + sqrt(Discriminant(a, b, c)))/2*a
    xMinus = (-b - sqrt(Discriminant(a, b, c)))/2*a
    return [xPlus, xMinus]

a = 1
b = -4
c = -12
d = QuadSolver(a, b, c)

print(d[0])
print(d[1])

```

Solving the quadratic in python. The discriminant is what's INSIDE the square root.

```

In [1]: run hw4pr3ex2.py
6.0
-2.0

```

Example outputs

```

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

int Discriminant(int a, int b, int c){
    int d;
    d = b*b - 4*a*c;

    return d;
}

int xPlus(int a, int b, int c){
    float xP;

    xP = (-b + sqrt(Discriminant(a, b, c)))/2*a;

    return xP;
}
int xMinus(int a, int b, int c){
    float xM;

    xM = (-b - sqrt(Discriminant(a, b, c)))/2*a;

    return xM;
}

int main() {
    int a;
    int b;
    int c;
    int xP_answ;
    int xM_answ;

    a = 1;
    b = -4;
    c = -12;

    xP_answ = xPlus(a, b, c);
    xM_answ = xMinus(a, b, c);

    printf("%d\n",xP_answ);
    printf("%d\n",xM_answ);

    return 0;
}

```

Same solver but in C

```

PS C:\Users\pamor\Desktop\hw week 4> ./a.exe
6
-2
PS C:\Users\pamor\Desktop\hw week 4>

```

Example outputs for C

This problem was simple enough as we already had done it in coral. Python was very intuitive and was quicker overall to program with.