

```
#-----#
#      Connor Seemann Seat 23 Lab 4.1      #
#-----#
```

```
# Program that will find the cost of your bagels in the morning
print("This program will find the cost of your bagels in the morning!")
```

```
ammount = int(input("How many bagels are ordered: ") )
cost = ammount * 0.60 if (ammount > 6) else ammount * 0.80
```

```
print("The cost of {bagels} is ${cost:.2f}".format(bagels=ammount, cost=cost) )
```

```
#-----#
#      Connor Seemann Seat 23 Lab 4.2      #
#-----#
```

```
from math import sqrt
```

```
# Program that will find the roots to second degree polynomials
print("This program will find the roots of a polynomial to the second degree")
```

```
term = [] #assigning the type of term as a list
term = eval( input("Enter a, b, c: ") ) # getting user input for the terms
```

```
a = term[0]; b = term[1]; c = term[2]
```

```
if ((b**2 - 4 * (a) * (c)) < 0): # checks to see if the discriminate is
    negative
```

```
    x1 = ( -b + sqrt( -1*(b**2 - 4 * a * c) ) ) / ( 2 * a ) # makes the
        discriminate positive and will add the i in the print statement
```

```
    x2 = ( -b - sqrt( -1*(b**2 - 4 * a * c) ) ) / ( 2 * a ) # makes the
        discriminate positive and will add the i in the print statement
```

```
if (x1 == x2):
```

```
    print("The roots are {x1:.2f}i".format(x1 = x1))
```

```
elif (x1 != x2):
```

```
    print("The roots are {x1:.2f}i, {x2:.2f}i".format(x1=x1, x2=x2))
```

```
elif ((b**2 - 4 * a * c) >= 0): # checks to see if the discriminate is positive
```

```
    x1 = ( -b + sqrt( b**2 - (4 * a * c) ) ) / ( 2 * a ) # finds the +
        discriminate
```

```
    x2 = ( -b - sqrt( b**2 - (4 * a * c) ) ) / ( 2 * a ) # finds the -
        discriminate
```

```
if (x1 == x2):
```

```
    print("The roots are {x1:.2f}".format(x1 = x1))
```

```
elif (x1 != x2):
```

```
    print("The roots are {x1:.2f}, {x2:.2f}".format(x1=x1, x2=x2))
```