**Explanation**

To give a brief explanation to go along with my flow chart, my code is simply 2 nested while loops nested in third while loop. The first nested while loop finds local minimum. If the next price in the price list is lower than the current price it takes that price as it is value. It then checks to see if the list is used up as a stop point, if not it stores that buy value and goes onto the next while loop to find a local maximum. Similarly, to the finding the minimum value if the next price is higher this time it will take the higher price until the next price is lower. It then stores that sell value and both the buy and sell values are used to calculate the profit value for that loop. The function will keep looping and adding each profit loop to the last until it reaches the end of the list of prices.

**Extra Credit tasks:**

1. **Consider how you would implement and test a low-frequency algorithmic trading program, i.e., one in which a human places trades based upon program recommendations.**

I suppose I would take some historic market data and use that as a starting point for the program to make recommendations. Using that data I would probably test it against the live market without making any purchases and see how it does.

1. **How might a trading program go about detecting another algorithmic agent acting in the market?**

I suppose people have been looking for patterns in the stock market for years. As hard as that would be I suppose computers could be exploited. If you know the times are days that certain programs make trades that would be a huge advantage of knowing when to buy and sell.