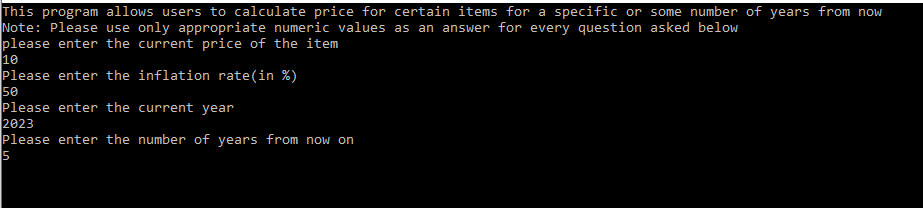
**Project 1 Question 1 Version 2 Documentation**

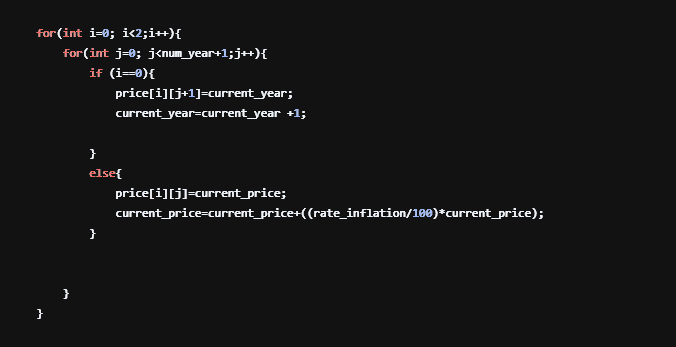
1. **Previous Versions**
   1. **The Aim of The Program:**The aim of the program is to predict the price of an item after or for certain years based on the inputs given by the user. Those inputs given by the user are: current price of the item, the current inflation rate, current year, number of years, choice number, and specific year.



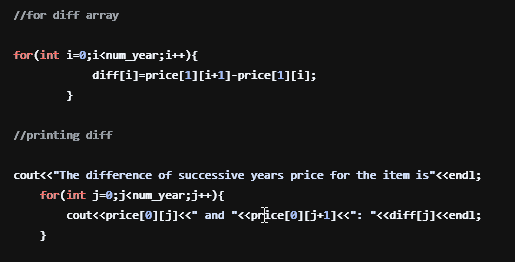
*Fig 1.1: Shows how the program takes input from a user*

* 1. **The Process:**

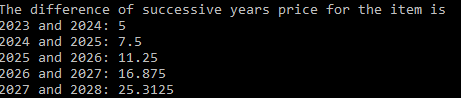
After the program takes the necessary inputs, it calculates the difference between the price of the item in successive two years and displays it to the user It calculates the difference of the price between each successive years using the difference between each values stored in successive indexes in price array (The price array is created to store the price of each year using the formula: current price+((inflation rate/100)\* current price)).



*Fig 1.2: Shows how the price array store the price of each year for certain period of years in the code*



*Fig 1.3: Shows the code that stores the differences of the price of successive years and display it to the user*

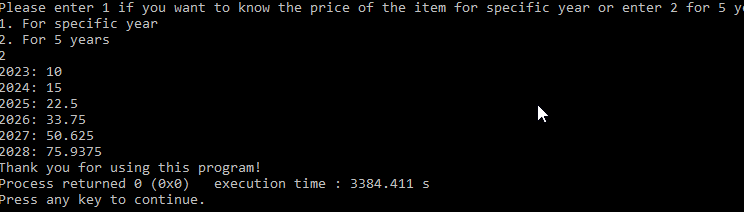
**

*Fig 1.4: Shows how the differences of each successive years is displayed on the screen*

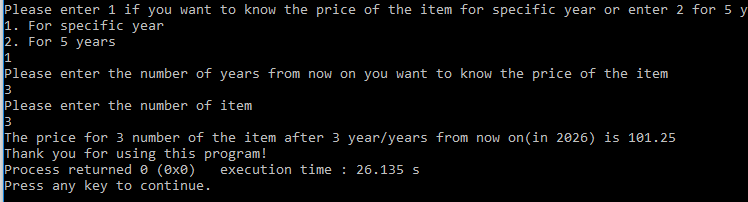
After calculating and displaying the differences of each successive years to the user, the program then prompts the user choose from two choices: the price of item for specific year or for certain period of years given by the user.



*Fig 1.5: Shows how the program prompts the user to choose*

**

*Fig 1.6: Shows what the program displays if the user enters choice number 2*

**

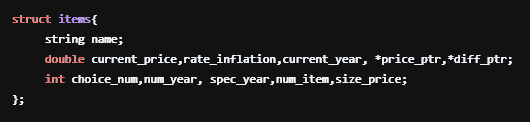
*Fig 1.7: Shows what the program displays if the user enters choice number 1*

1. **Project 1 Question 1 Version 2**
   1. **New Features Added:**

This updated version of Project 1 Question 1 has added and utilized the concepts of structure, functions, pointer, and header file in the previous versions which enabled the program to be more functional. The following points below describe the task of the newly added features in the code.

* + 1. **Structure**

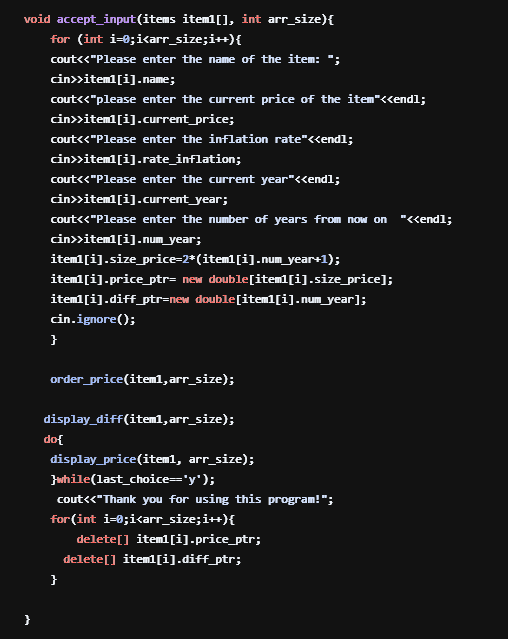
The structure in the code is used as a container to declare variables needed for a specific task as well as needed for storing the inputs listed and used in the previous versions of the project with one more added input, number of **types** of items.



*Fig 2.1: Shows how the structure stores and declares variables*

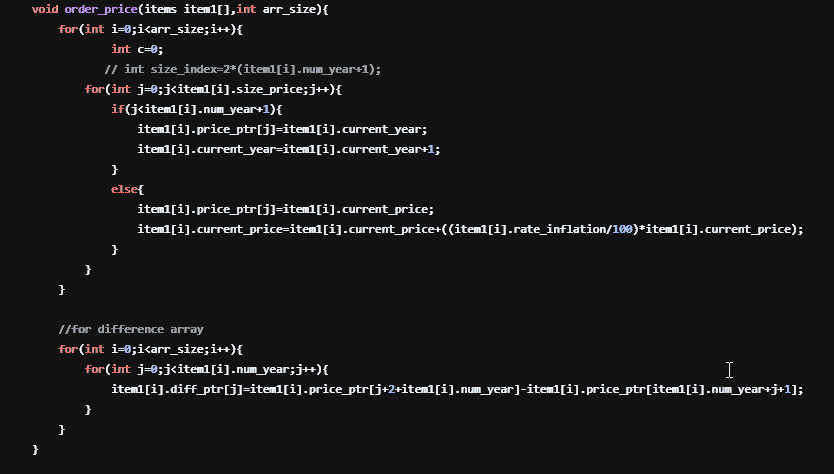
* + 1. **Functions**

This newer version of the program used 4 functions that perform specific task. The first one is the “accept\_input” function. It has two parameters and has no return type. It is used to accept the inputs given by the user. Refer Fig 1.1 to see how the console takes the inputs from the user. However, in addition to the previous versions input, the “accept\_input” functions display the new input to the user, which is not found in the previous versions.



*Fig 2.2: Shows how the “accept\_input” takes inputs from the user*

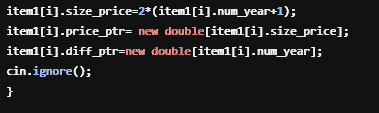
*Note: The “arr\_size” parameter used to accept the number of types of the items.*

The second function is called “order\_price”. It has two parameters and no return type. Its task is to store the inputs in arrays called price\_ptr and diff\_ptr. These arrays store the price for each successive years and differences of them respectively using the same logic as the previous versions.  *Fig 2.3: The “order\_price” function*

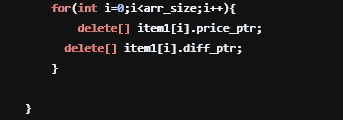
The third and the fourth functions task is to display information to the user. The former displays the difference of the prices of each successive years, whereas the later displays the price/s of the item for/after certain number of years entered by the user. Refer figures (Fig 1.4-1.7) to see how the outputs are displayed on the console.

* + 1. **Pointers**

In this version of the project, two pointers, namely “\*price\_ptr” and “\*diff\_ptr “, are used. They function as “diff” and “price” arrays in the previous versions. However, these are different in that they utilize the concept of dynamic memory management to store the differences of the prices in each successive years and prices of each year. Both the memory is allocated and deallocated in the “accept\_input” function.



*Fig 2.4: Shows how the memory is allocated dynamically in the program*



*Fig 2.5: Shows how the memory is deallocated in the program*

* 1. **Advantages of this Version:**
* The use of structure enabled the program to easily process the prices and differences for different number of types of items that was not included in the previous versions.
* The use of functions enables to debug as well as add new features easily
* The use of functions and header file increased the readability of the program
* The use of dynamic memory management system
* It loops over the results as long as the user wants to do so