LAB 1

**How it implements the problem statement:**

Given the number of items, names and prices they do the following which computes the average price of the item if “peas” are included. Also it prints the items in reverse order along with their prices. The code implements the problem statement with two arrays which are item and price. A loop is used to input the names and prices and compute the average price. Another loop is also used in order to print the items in reverse order along with their prices.

**CODE:**

for (int i = 0; i < numitems; i++)

{

System.out.println("Enter item name: ");

item[i] = input.nextLine();

String red = input.nextLine();

System.out.println("Enter item price: ");

price[i] = input.nextDouble();

if ( item[i].equalsIgnoreCase("Peas"))

{

average = price[i] / numitems;

total=true;

}

}

**Analysis:**

This loop prints two prompts and reads two inputs. This is dependent on the amount of iterations you want the inputs to have. The total time spent in this loop is directly proportional to the number of iterations which is n items. The main focus of this loop is to allocate “items” and “price” inputs into each array and to store them.

**Time complexity:**

=n+1

=n+1

=2n^2

t(n)= O(2n^2)

=O(n^2)

**CODE:**

for (int j = numitems - 1; j >= 0; j--)

{

System.out.println(item[j] + " " + price[j]);

}

**Analysis:**

The following code gives both arrays “ item” and “price” in reverse order. Which led them to print the items and their prices starting for the last item and price that were input which are now going to be first. Numitems-1 ensures that the loop outputs in the reverse order. It outputs in reverse order until j is equal to 0 therefore stopping the loop and finally leading to the end of the program.

**Time complexity:**

=n-1

=n

=2n-1

t(n)=O(n)