AP® Computer Science A
2002 Sample Student Responses

In writing `ChangePrices`, you may call any of the public member functions of the `GroceryStore` class. Assume the member functions work as specified.

Complete free function `ChangePrices` below.

```
void ChangePrices(GroceryStore & store, istream & input)
// precondition:   input is open for reading;
//                 each line consists of a valid one word item name
//                 and a valid price
// postcondition:  changes the prices of items in store using names and
//                 new prices from input
{
    apstring name;
    double price;
    while (input >> name >> price)
    {
        store.SetPrice(name, price);
    }
}
```

GO ON TO THE NEXT PAGE

Complete free function `BargainItem` below.

```
apstring BargainItem(const GroceryStore & store, char category)
// postcondition: returns the name of an item whose unit price
//                is the lowest in the specified category;
//                if no items in the specified category, returns "none"
{
    double price = 0, low = 0;
    apvector <apstring> list = store.GetItems(category);
    apstring lowest;
    if( list.length == 0)
        return "none";

    for(int i = 0; i < list.length(); i++)
    {
        price = store.GetPrice(list[i]) / store.GetSize(list[i]);
        if(price < low || low == 0)
        {
            low = price;
            lowest = list[i];
        }
    }
    return lowest;
}
```

In writing `ChangePrices`, you may call any of the public member functions of the `GroceryStore` class. Assume the member functions work as specified.

Complete free function `ChangePrices` below.

```
void ChangePrices(GroceryStore & store, istream & input)
// precondition:  input is open for reading;
//                each line consists of a valid one word item name
//                and a valid price
// postcondition: changes the prices of items in store using names and
//                new prices from input
```

```
{
    While (input >> item name >> price)
        Store. setPrice(item name, price)
}
```

Complete free function `BargainItem` below.

```
apstring BargainItem(const GroceryStore & store, char category)
// postcondition: returns the name of an item whose unit price
//                is the lowest in the specified category;
//                if no items in the specified category, returns "none"
```

```
{
    int low = 3200, count;
    apstring n;
    apvector <apstring> list = get items (category);
    for (count = 0; count < list.length(); count++)
    {
        if (store.GetPrice (list [count]) / store.getsize(list [count]) < low)
        {
            low = store.getprice (list [count]) / store.getsize(list[count]);
            n = list [count];
        }
    }
    return n;
}
```

In writing `ChangePrices`, you may call any of the public member functions of the `GroceryStore` class. Assume the member functions work as specified.

Complete free function `ChangePrices` below.

```
void ChangePrices(GroceryStore & store, istream & input)
// precondition:  input is open for reading;
//                each line consists of a valid one word item name
//                and a valid price
// postcondition: changes the prices of items in store using names and
//                new prices from input
{
    string currentname;
    double currentprice;


    while (input >> currentname)
    {
        input >> currentprice;
        SetPrice(currentname, currentprice);
    }

    return;

}
```

Complete free function `BargainItem` below.

```
apstring BargainItem(const GroceryStore & store, char category)
// postcondition: returns the name of an item whose unit price
//                is the lowest in the specified category;
//                if no items in the specified category, returns "none"
{
```

```
apstring currentbargain ("none");

apvector <apstring> categoryitems = GetItems (category);
int x;
double lowest, current;
if (categoryitems.length())
   {lowest = GetPrice(categoryitems[0]);
    currentbargain = categoryitems[0]; }
for(x=1; x< categoryitems.length(); x++)
   {
     current = GetPrice(categoryitems[x]);
     if (current < lowest)
        {
        lowest = current;
        current bargain = categoryitems[x];
        }
   }

return currentbargain;

}
```