Homework #3 Shop List Test Plan
Yingtong Zhou

1) **Inputs and Outputs**
    a. Input: name, price, priority, quantity
    b. Output: a sorted list with item descriptions, a final display of purchased and unpurchased items with item descriptions
2) **Define Classes, Methods and Data structures**
    - ShoppingDemo (Main Class)
        - Create a ItemDesc list to store a series of ItemDesc objects
        - Read inputs from the user, check if there are duplicate items, check if total price is larger than $100
        - Allow user to specify the priorities for each item
        - Display the sorted shop list based on priorities
        - Display a final list of what was purchased, its price and what wasn't purchased
    - ItemDesc
        - Private instance variables: name, price, priority, quantity
        - Constructor to initialize the item
        - Getter and setter

3) **UML**

ItemDesc
- **name: String** - **price: double** - **priority: int** - **quantity: int**
+ getName(): String **+ setName(String name): void** **+ getPrice(): double** **+ setPrice(double price): void** **+ getPriority(): int** **+ setPriority(int priority) : void** **+ getQuantity(): int** **+ setQuantity(int quantity): void**

ShoppingDemo
- **size: int**
+ main(String[]): void **+ bubbleSort(int[], int): void** **+ equals(Object): boolean** **+ purchaseItem(ItemDesc[], double)**

4) Pseudo Code

```
Size = 7;
For (i=0; i<size; i++)
{
    Iterate the partially filled array (length: i), check if input item exists. If exists, let user reinput.

    for (j=0; j<i; j++) {
        If (name.equals(input name)) {
            Reinput name
        }

    Iterate the array, add the price.

    sum += price * quantity
 }

 Read the integer user inputs, and let user change the item he chose. Repeat this step until total
sum >= 100.
 while (sum<100)
 {
    flag = true
    while flag {
        switch (i)
        case 1,2,3,4,5,6,7 -> {
            sum -= original sum of the item
            reset price
            sum += new sum of the item
            flag = false
            break
        }
    }
 Create a new array to store the priorities.
 Sort the array.
 If (priority == input priority) {
    Set the item desc of new array}
 Display final list.
```