Assignment 5 - Postal Packages

AllyBaba Shipping Service ships packages. Packages are accepted for shipping subject to the following restrictions:

Shipping Requirements

Package Size/Weight Requirements

- The package weight must not exceed 50 pounds.
- The package must not exceed 6 feet in length, width, or height.
- The girth of the package must not exceed 10 feet. The girth is the circumference around the two smallest sides of the package. If *side1*, *side2*, and *side3* are the lengths of the three sides, and *largest* is the largest of the three sides, then the *girth* can be calculated using:
  girth = 2 * (*side1* + *side2* + *side3* - *largest*)

Shipping Location Requirements

- For Texas, there is no state charge
- For out-of-state, there is a $35.00 service charge
- For out-of-country, there is $40.00 service charge

Shipping Charges

Weight	Shipping Charge
1	1.50
2	2.10
3	4.00
5	6.75
7	9.90
10	14.95
13	19.40
16	24.20
20	27.30
25	31.90
30	38.50
35	43.50
40	44.80
45	47.40
50	55.20

Transaction Requirement

1. For each transaction, the user will select (T)exas; (O)ut of state; (F)oreign
2. For each transaction (package to be shipped), the user should enter the package weight followed by the 3 package dimensions in any order. The weight should be specified as a whole number of pounds and the dimensions are specified as a whole number of inches.
3. To end the program, the user should enter a weight of X. When the user enters X to end the program, they should not have to enter values for the 3 package dimensions. That is, when

the user is ready to end the program, they should only have to enter the weight:

Input Validation

1. Check to be sure that the user enters positive numbers for the package weight and dimensions (weight and dimensions must be larger than zero). For transactions with invalid weight or dimensions, print an error message and skip the transaction.
2. The shipping charge is based on the shipping charge table. This table may be represented in your program as parallel arrays (two one-dimensional arrays), one for the weight and one for the shipping charge. Alternatively, you may define a struct or simple class to represent one table entry (one weight and cost). Then the table would be an array of structs or an array of objects.
   Note: Do not use a two-dimensional array to store the weights and costs. The weights need to be stored as integers, and the costs need to be stored as floating-point values. So they cannot be mixed in one array.
   You can initialize the array elements in the array declarations using the values from the table.
3. To determine the shipping charge, search the weight array for the package weight and then use the corresponding element from the shipping charge array.

   For example, the shipping charge for a 3 pound package would be $4.00. If the package weight falls between the weights in the weight table, use the larger weight. For example, the shipping charge for a 4 pound package would be 6.75.
4. **Do not hard code these values into your program code.** For example, **you should NOT have code like:**

```
if ( packageWeight == 4 || packageWeight == 5)
   shippingCost = 6.75;
```

Program 5 Hint

1. You can use a standard sequential search to search the array for the package weight -- you only need a very minor modification.
2. The standard sequential search stops when an exact match for the search item is found in the array. A standard search will not work for this problem since all possible weights (1 - 50) are not stored in the array. For example, a 6 pound package would not have a matching weight -- the shipping charge would be the same as for a 7 pound package.
3. The required modification is simple. You need a search that stops when the package weight is less than or equal to the weight in the array.

Output

Note that you should be able to write this program with only two loops, a transaction processing loop and a cost search loop. You do not need an input validation loop. If the package input is invalid, just print an error message and skip the normal package processing.
I want you to input the package information, process the package and output some information about the package in a single loop (the transaction processing loop).
Your program output should look similar to the example below.
Note: Do not store the package information in an array or vector. If you process a lot of packages, you will eventually run out of space in the array. If you use a vector, as you process more transactions you will eat up more and more memory. This program should be designed to run for long periods of time without running out of memory.

For each transaction, print:

- The destination
- The transaction number (start counting with 1)
- Whether the package was accepted or rejected
- The package weight
- The cost for shipping (if applicable)

When the program ends, print the number of packages accepted for shipping, the number of packages rejected and the total charge. Transactions that contain invalid input should not be counted (see Input Validation section).

Additional Requirements:

1. Do not use global variables in any assignment. A global variable is a variable that is declared outside any function. It is okay to use global constants.
2. You must use multiple functions in this program. Do not write the program as one large main function.
3. Do not store the transaction information in arrays or vectors. When you store transactions in arrays, you set a limit on the number of transactions your program can process. For this program, you should be able to read and process transactions one at a time.

Example
The screen dialog should look similar to this (user input is shown in **bold**):
WELCOME TO ALLYBABA SHIPPING SERVICE

Menu

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it
Enter package weight and 3 dimensions.
Enter 0 to quit.

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it **T**
Enter package weight **1**
Enter side1 **2**
Enter side2 **3**
Enter side3 **3**

Transaction #1

Location	Texas
Status	Accepted
Weight	1 lb
Cost	$1.50

Menu

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it
Enter package weight and 3 dimensions.
Enter 0 to quit.

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it **T**
Enter package weight **7**
Enter side1 **4**
Enter side2 **2**
Enter side3 **3**

Transaction #2

Location	Texas
Status	Accepted
Weight	7 lbs
Cost	$9.90

Menu

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it
Enter package weight and 3 dimensions.
Enter 0 to quit.

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it **T**
Enter package weight **21**
Enter side112

Enter side2 **15**
Enter side3 **12**

Transaction #3
Location Texas
Status Accepted
Weight 21 lbs
Cost $31.90

Menu

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it
Enter package weight and 3 dimensions.
Enter 0 to quit.

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it **T**
Enter package weight 61
Enter side1 **24**
Enter side2 **40**
Enter side3 **20**

Transaction #4
Location Texas
Status Rejected
Weight 61 lbs
Cost N/A

Menu

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it
Enter package weight and 3 dimensions.
Enter 0 to quit.

Enter Location - (T)exas; (O)ut of state; (F)oreign (X)it **X**

Number of accepted packages 3
Number of rejected packages
Total Cost $XXX.XX