Homework Exception

1. Write a program that converts time from numerical hour-minute-second format to a formatted string (for example, 14:30:45 corresponds to "14 hours, 30 minutes, and 45 seconds"). You will define three exception classes: `HourError`, `MinuteError`, and `SecondError`. If the user enters anything other than a legal hour number (integers from 0 to 23), then your program will throw and catch an `HourError`. Similarly, if the user enters anything other than a valid minute number (integers from 0 to 59), then your program will throw and catch a `MinuteError`. Finally, if the user enters anything other than a valid second number (integers from 0 to 59), then your program will throw and catch a `SecondError`. Your driver program should test a few normal and two exceptional cases. Followings are test data:

```
int testCases[][3] = {
       {14, 30, 45},   // Valid
       {23, 59, 59},   // Valid
       {24, 0, 0},    // Invalid hour
       {12, 60, 0},    // Invalid minute
       {12, 30, 60},   // Invalid second
       {25, 60, 20},   // Invalid hour and minute
       {26, 70, 60},   // Invalid hour, minute and second
```

2. The following code uses two arrays, one to store products and another to store product IDs (a better organization would be to use a single array of a class or struct, but that is not the subject of this Programming Project). The function `getProductID` takes as input the two arrays, the length of the arrays, and a target product to search for. It then loops through the product name array; if a match is found, it returns the corresponding product ID:

```
    int getProductID(int ids[], string names[], int numProducts, string
    target)
    {
        for (int i = 0; i < numProducts; i++)
        {
            if (names[i] == target)
                return ids[i];
        }
        return -1; // Not found
    }
```

```
    int main() // Sample code to test the getProductID function
    {
        int productIds[] = {4, 5, 8, 10, 13};
        string products[] = {"computer", "flash drive", "mouse", "printer",
                             "camera"};

        cout << getProductID(productIds, products, 5, "mouse") << endl;
        cout << getProductID(productIds, products, 5, "camera") << endl;
        cout << getProductID(productIds, products, 5, "laptop") << endl;

        return 0;
}
```

One problem with the implementation of the `getProductID` function is that it returns the special error code of -1 if the target name is not found. The caller might ignore the -1, or later we might actually want to have -1 as a valid product ID number. Rewrite the program so that it throws an appropriate exception when a product is not found instead of returning -1.

Requirements:

(1) Define a custom exception class `ProductNotFoundException` that inherits from `std::exception`.

(2) Modify the `getProductID` function to throw `ProductNotFoundException` when the target product is not found.

(3) Update the main function to handle the exception using a try-catch block.