CPP Problem Design Example

Subject：Template Binary Search

Contributor: 廖宣瑋, 陳俊儒, 林承達

Main testing concept: Templates

Basics

□ C++ BASICS
□ FLOW OF CONTROL
□ FUNCTION BASICS
□ PARAMETERS AND OVERLOADING
□ ARRAYS
□ STRUCTURES AND CLASSES
□ CONSTRUCTORS AND OTHER TOOLS
□ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES
□ STRINGS
□ POINTERS AND DYNAMIC ARRAYS

Functions

□ SEPARATE COMPILATION AND NAMESPACES
□ STREAMS AND FILE I/O
□ RECURSION
□ INHERITANCE
□ POLYMORPHISM AND VIRTUAL FUNCTIONS
■ TEMPLATES
□ LINKED DATA STRUCTURES
□ EXCEPTION HANDLING
□ STANDARD TEMPLATE LIBRARY
□ PATTERNS AND UML

Description:

Please implement Binary Search using template, and provide iterative and recursive versions. The iterative version of the function should follow this format ItrBinarySearch(const T a[], int first , int last, T key, bool &found, int &location), recursive version of the function please follow this format RecBinarySearch (const T a[], int first, int last, T key, bool &found, int &location), both functions please support int, string and double types. Please refer to the following main. When we score, we will use our main to replace your main, so be sure to follow the above format.

```
int main()
{
	const int ARRAY_SIZE = 8;
	const int finalIndex = ARRAY_SIZE - 1;

	int i;
	int a[] = { 1, 2, 3, 4, 10, 25, 19, 100 };
	// Test int
	cout << "\nArray contains:\n";
	for (i = 0; i < ARRAY_SIZE; i++)
	{
		cout << a[i] << " ";
	}
	cout << endl;
	int keyInt, location;
	bool found;
	cout << "Enter number to be located: ";
	cin >> keyInt;

	cout << "Testing Template Iterative Binary Search\n";
	ItrBinarySearch(a, 0, finalIndex, keyInt, found, location);
	if (found)
		cout << keyInt << " is in index location " << location << endl;
```

```
    else
        cout << keyInt << " is not in the array." << endl;


    cout << "Testing Template Recursive Binary Search\n";
    RecBinarySearch(a, 0, finalIndex, keyInt, found, location);
    if (found)
        cout << keyInt << " is in index location " << location << endl;
    else
        cout << keyInt << " is not in the array." << endl;

    // Test string
    string b[] = { "aa", "ab", "ah", "bd", "be", "cc", "fe", "zk" };
    string keyString;
    cout << "\nArray contains:\n";
    for (i = 0; i < ARRAY_SIZE; i++)
    {
        cout << b[i] << " ";
    }
    cout << endl;
    cout << "Enter number to be located: ";
    cin >> keyString;

    cout << "Testing Template Iterative Binary Search\n";
    ItrBinarySearch(b, 0, finalIndex, keyString, found, location);
    if (found)
        cout << keyString << " is in index location " << location << endl;
    else
        cout << keyString << " is not in the array." << endl;


    cout << "Testing Template Recursive Binary Search\n";
    RecBinarySearch(b, 0, finalIndex, keyString, found, location);
    if (found)
        cout << keyString << " is in index location " << location << endl;
    else
        cout << keyString << " is not in the array." << endl;

    // Test double
    double c[] = { 0.3 , 5.6 , 7.8 , 10.9 , 123.5 , 150.1 , 197.1 ,
2019.2 };
    double keyDouble;
    cout << "\nArray contains:\n";
    for (i = 0; i < ARRAY_SIZE; i++)
    {
        cout << c[i] << " ";
    }
    cout << endl;
    cout << "Enter number to be located: ";
    cin >> keyDouble;

    cout << "Testing Template Iterative Binary Search\n";
    ItrBinarySearch(c, 0, finalIndex, keyDouble, found, location);
    if (found)
        cout << keyDouble << " is in index location " << location << endl;
    else
        cout << keyDouble << " is not in the array." << endl;


    cout << "Testing Template Recursive Binary Search\n";
    RecBinarySearch(c, 0, finalIndex, keyDouble, found, location);
    if (found)
        cout << keyDouble << " is in index location " << location << endl;
    else
        cout << keyDouble << " is not in the array." << endl;
    return 0;
}
```

Input:

Please enter the number of times to be tested N, and then enter N sets of int, string, and double, respectively.
We will use our main to replace your main, so be sure to follow the above format.

Output:

Please output all the elements in the array first, and then output the index value of the binary search in the iterative and recursive versions, respectively, e.g. “(value) is in index location (index)”, if the value is not in the array, then output “(value) is not in the array.”.

Sample Input / Output：

Sample Input	Sample Output
3 1 aa 0.3 100 zk 2019.2 5 gg 2018.2	 Array contains: 1 2 3 4 10 25 29 100 Enter number to be located: Testing Template Iterative Binary Search 1 is in index location 0 Testing Template Recursive Binary Search 1 is in index location 0 Array contains: aa ab ah bd be cc fe zk Enter number to be located: Testing Template Iterative Binary Search aa is in index location 0 Testing Template Recursive Binary Search

```
aa is in index location 0

Array contains:
0.3 5.6 7.8 10.9 123.5 150.1 197.1 2019.2
Enter number to be located:
Testing Template Iterative Binary Search
0.3 is in index location 0
Testing Template Recursive Binary Search
0.3 is in index location 0

Array contains:
1 2 3 4 10 25 29 100
Enter number to be located:
Testing Template Iterative Binary Search
100 is in index location 7
Testing Template Recursive Binary Search
100 is in index location 7

Array contains:
aa ab ah bd be cc fe zk
Enter number to be located:
Testing Template Iterative Binary Search
zk is in index location 7
Testing Template Recursive Binary Search
zk is in index location 7

Array contains:
0.3 5.6 7.8 10.9 123.5 150.1 197.1 2019.2
Enter number to be located:
Testing Template Iterative Binary Search
2019.2 is in index location 7
Testing Template Recursive Binary Search
2019.2 is in index location 7

Array contains:
1 2 3 4 10 25 29 100
Enter number to be located:
Testing Template Iterative Binary Search
5 is not in the array.
Testing Template Recursive Binary Search
5 is not in the array.

Array contains:
aa ab ah bd be cc fe zk
Enter number to be located:
Testing Template Iterative Binary Search
gg is not in the array.
Testing Template Recursive Binary Search
gg is not in the array.
```

```
Array contains:
0.3 5.6 7.8 10.9 123.5 150.1 197.1 2019.2
Enter number to be located:
Testing Template Iterative Binary Search
2018.2 is not in the array.
Testing Template Recursive Binary Search
2018.2 is not in the array.
```

□ Eazy,Only basic programming syntax and structure are required.

■ Medium,Multiple programming grammars and structures are required.

□ Hard,Need to use multiple program structures or more complex data types.

Expected solving time:

30 minutes

Other notes:

The array elements to be searched must be in ascending order.