SOLUTION TO EXERCISE 8 - ANALYSIS PROBLEM

The following documentation provides one solution for Array Analysis Exercise 8, which provided partial documentation and requested the development of the remaining items below. Note: All italicized text below serves only to clarify the documentation for students, but would not actually appear in the real documentation.

PROBLEM STATEMENT (Given in Assignment)

SAMPLE SOFTCOPY (Given in Assignment)

SYMBOLIC CONSTANT LIST

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>DESCRIPTION</th>
<th>DATA TYPE</th>
<th>VALUE</th>
<th>USAGE</th>
<th>DESTINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIZE</td>
<td>Quantity of array elements</td>
<td>Integer</td>
<td>20</td>
<td>Tested</td>
<td>---</td>
</tr>
</tbody>
</table>

VARIABLE LIST (Given in Assignment)

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>DESCRIPTION</th>
<th>DATA TYPE</th>
<th>SOURCE</th>
<th>USAGE</th>
<th>DESTINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRAYS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Array of SIZE integers</td>
<td>Integer</td>
<td>Given *</td>
<td>Tested</td>
<td>---</td>
</tr>
<tr>
<td>SCALARS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET</td>
<td>Target value of search</td>
<td>Integer</td>
<td>Keyboard</td>
<td>Tested</td>
<td>Screen</td>
</tr>
<tr>
<td>I</td>
<td>Index for Array N</td>
<td>Integer</td>
<td>Set to 0</td>
<td>Loop Index</td>
<td>---</td>
</tr>
<tr>
<td>QTY</td>
<td>Quantity of hits on TARGET in array N</td>
<td>Integer</td>
<td>Set to 0</td>
<td>Accumulates 1 for each hit</td>
<td>Screen</td>
</tr>
</tbody>
</table>

* The initial analysis used the following array values (in ascending order by subscript):
  25  68  15  35  84  56  8  52  27  7  53  15  8  42  32  1  5  8  41  23

ALGORITHM/FLOWCHART

A flowchart is a graphic algorithm that helps to illustrate the flow of control of a program as the processor would advance from one step to another, often in simple sequential order, but sometimes following more complex structures. In this case the processor must use its ability to compare and recognize relationships between data such as greater, less than, or equal to, and make a decision about which step(s) to process next. The flowchart in this solution shows a selection structure nested inside a leading decision loop. For more information about what each shape below represents, view the web page about Flowcharting Symbols & Guidelines.

(See next page)
The desk check below was done using the array values given in the assignment (in ascending order by subscript):

25 68 15 35 84 56 8 52 27 7 53 15 8 42 32 1 5 8 41 23
**DATA TRACING CHART**

Data Tracing Charts are used to document what would be happening in the computer's memory during the execution of the steps described in an algorithm. A column is provided for each variable in your analysis and for each condition being tested (in a diamond shape). The array values are not changed by this program. As such there was no need to trace them. So no columns have been provided showing their values. In programs where array values do change, a column should be provided for each element in the array.

<table>
<thead>
<tr>
<th>TARGET</th>
<th>QTY</th>
<th>I</th>
<th>I &lt; SIZE</th>
<th>N₁ = TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>TARGET</th>
<th>QTY</th>
<th>I</th>
<th>I &lt; SIZE</th>
<th>N₁ = TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td></td>
<td></td>
<td>True</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>18</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td></td>
<td></td>
<td>False</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td>False</td>
</tr>
</tbody>
</table>

TEST SOFTCOPY

The numerals on the left of the sample output below are there for the analyst's reference only and will not appear on the screen. Bracketed items indicate values entered by the user.

1 ARRAY SCANNER
2 ===========
3
4 This program will Scan an array of stored integers seeking
5 a specific value and display a message indicating how many
6 times (if any) that value was found in the array.
7
8 Value to seek? 8
9
10 The value 8 was found 3 times in the array.
SOURCE CODE FOR ASSIGNMENT 8

The following source code was not assigned for Array Analysis Assignment 8, but has been included here to provide a complete example of the program's development.

/***************************************************
* Program: a8.cpp - Assignment 8 - Coding Solution *
***************************************************/
#include <iostream>    /* Include header to allow use of console functions */
using namespace std;
#include <iomanip>     /* Include header to allow use of stream manipulators */
#define  SIZE  20      /* Size of the array */

int main()
{
    /* Array Declaration - An array of known integers */
    int N[]={25,68,15,35,84,56,8,52,27,7,53,15,8,42,32,1,5,8,41,23};

    /* Scalar Declaration */
    int TARGET,    /* Target value of search */
        I,        /* Index for array N */
        QTY;      /* Quantity of hits on TARGET in array N */

    /* Process */
    cout << "ARRAY SCANNER\n";
    cout << "=============\n";
    cout << "This program will Scan an array of stored integers seeking\n";
    cout << "a specific value and display a message indicating how many\n";
    cout << "times (if any) that value was found in the array.\n\n";
    cout << "Value to seek? ";
    cin >> TARGET;
    QTY = 0;
    for (I=0; I<SIZE; I=I+1)
        if ( N[I] == TARGET ) QTY = QTY + 1;
    cout << "\nThe value " <<TARGET<< " was found " <<QTY<< " times in the array.\n";
    return 0;  /* Send a null error code to the parent process */
}