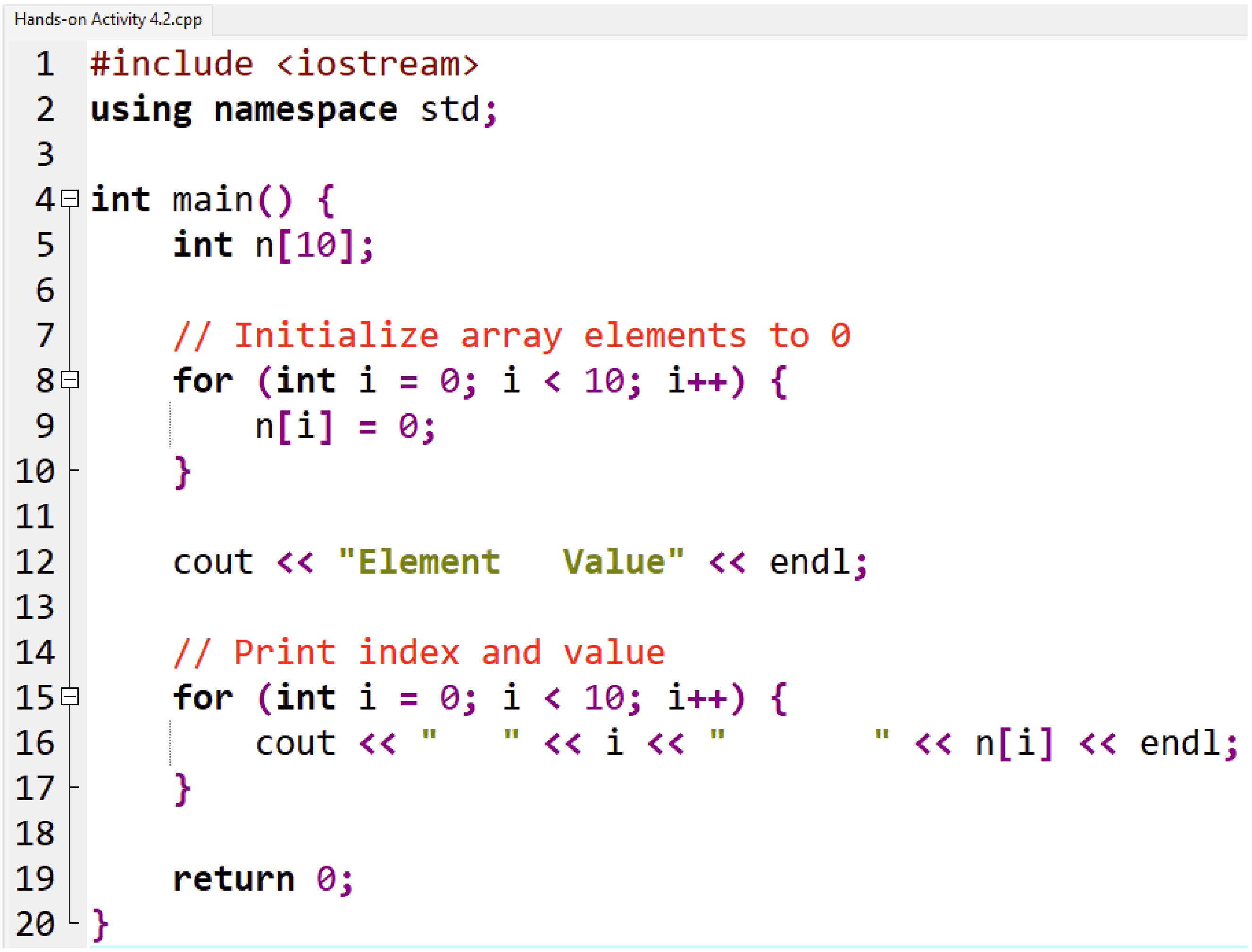
Hands-on Activity 4.2	
Arrays	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:** September 11, 2025
Section: CPE11S1	**Date Submitted:** September , 2025
Name(s): Mendoza, Nathaniel B.	**Instructor: Engr. Jimlord M. Quejado**
6. Output	

1. **Code:**



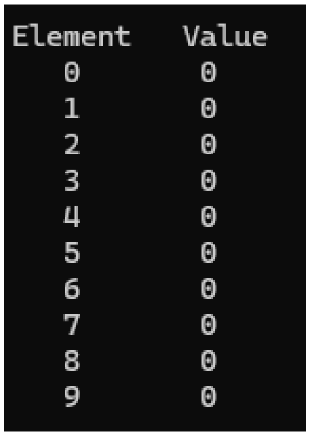
```cpp
#include <iostream>
using namespace std;

int main() {
    int n[10];

    // Initialize array elements to 0
    for (int i = 0; i < 10; i++) {
        n[i] = 0;
    }

    cout << "Element   Value" << endl;

    // Print index and value
    for (int i = 0; i < 10; i++) {
        cout << "   " << i << "       " << n[i] << endl;
    }

    return 0;
}
```

Code Explanation :

- The code concluded a #include <iostream> or Input/Output stream which gives access to code cout that's used for this code.
- using namespace std; has also been utilized for the code to shorten the coding.
- The “int main () {}” is the main function where the program starts running.
- int n[10] means the number of sets of data. Which is equal to 10, its counting starts from 0 to 9.
- Array starts at 0, so it is initialized at equals to 0.
- for(int i = 0; i < 10; i++) means, it will start incrementing a number of elements until it is 10,
- The expected value is all 0, so that it will be printed as cout<<” “<<i; since i=0.
- For printing its elements the expected values start from 0 to 9, so that you will apply n[i] so that it will keep incrementing or 0 keeps getting added by 1 until it is 10. That’s why it was written as <<” “<< n[1];

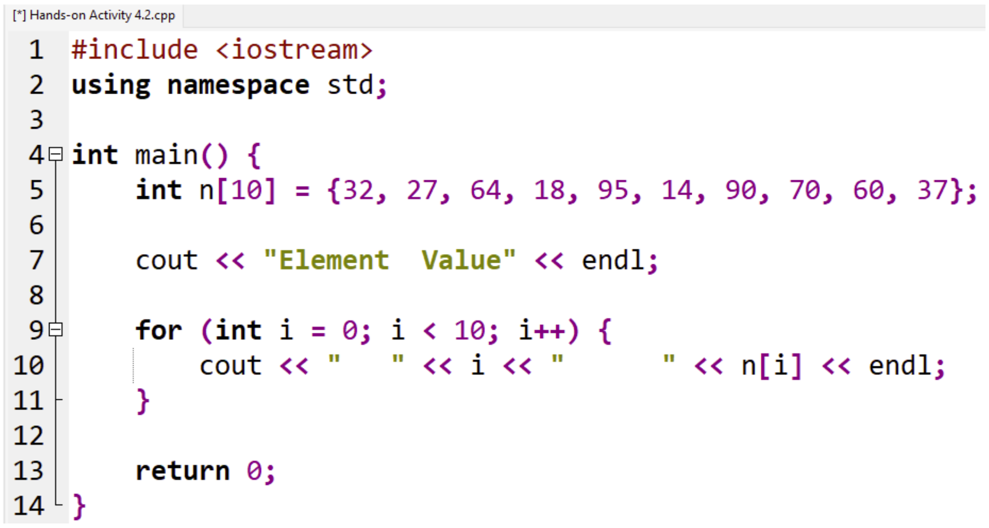
- return 0; } ends the program.

Code Output:



```
Element   Value
   0        0
   1        0
   2        0
   3        0
   4        0
   5        0
   6        0
   7        0
   8        0
   9        0
```

2. **Code:**



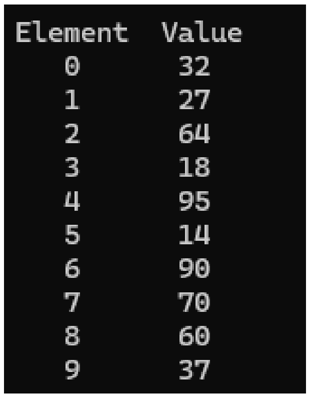
[*] Hands-on Activity 4.2.cpp

```cpp
#include <iostream>
using namespace std;

int main() {
    int n[10] = {32, 27, 64, 18, 95, 14, 90, 70, 60, 37};

    cout << "Element  Value" << endl;

    for (int i = 0; i < 10; i++) {
        cout << "   " << i << "       " << n[i] << endl;
    }

    return 0;
}
```

Code Explanation:

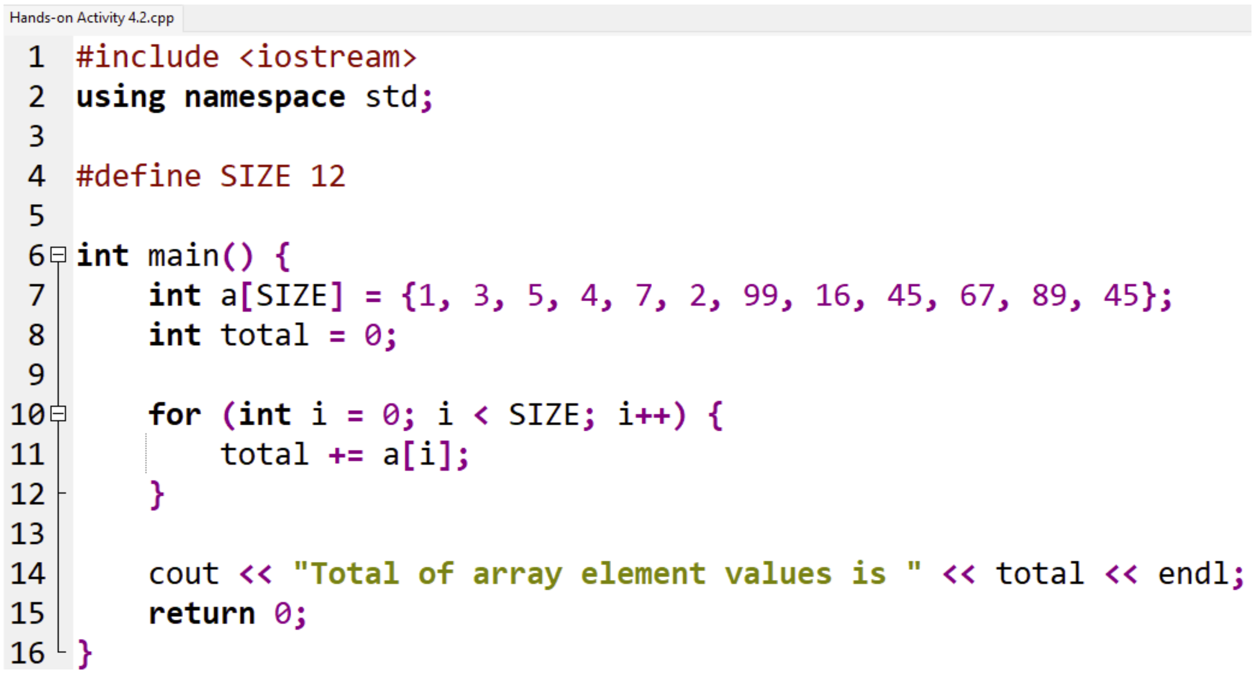
- The code concluded a #include <iostream> or Input/Output stream which gives access to code cout that's used for this code.
- using namespace std; has also been utilized for the code to shorten the coding.
- The “int main () {}” is the main function where the program starts running.
- int n[10] shows that the array has a set of numbers that consist of 10 total sizes.
- The expected values consist of an “Element” and a “Value” that is why it is printed.
- As for the for loop, the expected values of Element need to keep increasing until its 10, so that the code used incrementation for it to increase until it is equal to 10. As for the value, the print shows n[i] so that it will print all the given set of data from the array.
- return 0; } ends the program.

Code Output:



```
Element  Value
   0       32
   1       27
   2       64
   3       18
   4       95
   5       14
   6       90
   7       70
   8       60
   9       37
```

3. Code



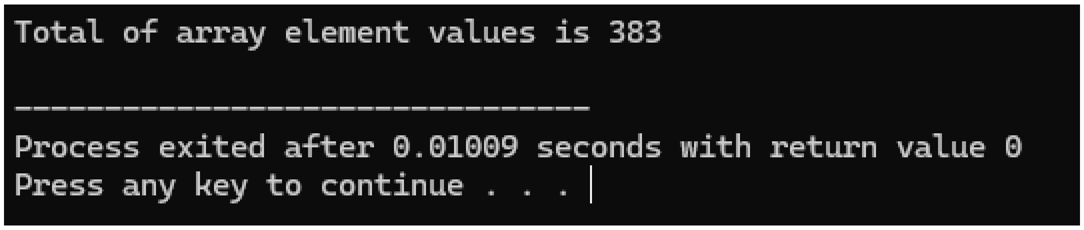
Hands-on Activity 4.2.cpp

```cpp
#include <iostream>
using namespace std;

#define SIZE 12

int main() {
    int a[SIZE] = {1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45};
    int total = 0;

    for (int i = 0; i < SIZE; i++) {
        total += a[i];
    }

    cout << "Total of array element values is " << total << endl;
    return 0;
}
```

Code Explanation:

- The code concluded a #include <iostream> or Input/Output stream which gives access to code cout that's used for this code.
- using namespace std; has also been utilized for the code to shorten the coding.
- The “int main () {}” is the main function where the program starts running.
- The code defines a total number of 12
- The array used its size to implement a set number of datas.
- The code implemented that the total is equal to 0 as there was nothing that happened yet.
- Used for loop to keep increasing the i until it is less than the given size, which is 12. And it will keep incrementing and keep getting the sum of the array until it ends.
- The last part prints its total sum from the given array.
- return 0; } ends the program.

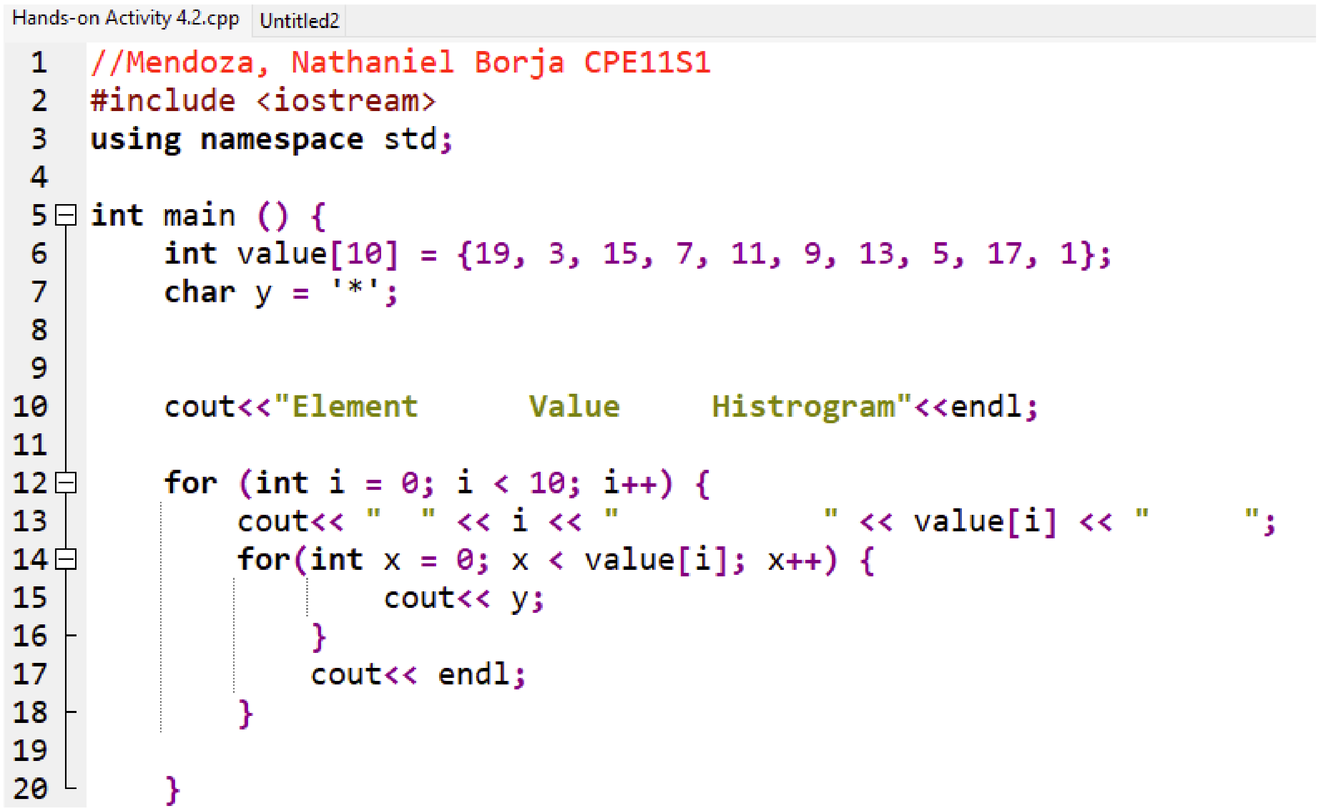
Code Output:



```
Total of array element values is 383

--------------------------------
Process exited after 0.01009 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

1. **Code:**

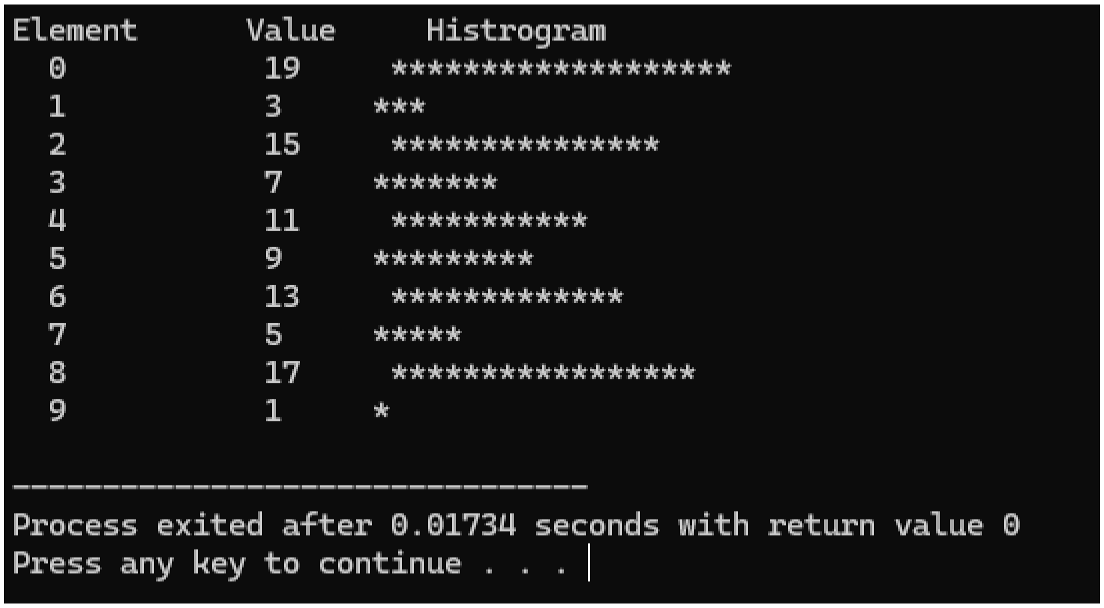


```cpp
//Mendoza, Nathaniel Borja CPE11S1
#include <iostream>
using namespace std;

int main () {
    int value[10] = {19, 3, 15, 7, 11, 9, 13, 5, 17, 1};
    char y = '*';


    cout<<"Element      Value     Histrogram"<<endl;

    for (int i = 0; i < 10; i++) {
        cout<< "  " << i << "           " << value[i] << "     ";
        for(int x = 0; x < value[i]; x++) {
                cout<< y;
            }
            cout<< endl;
        }

    }
```

Code Explanation:

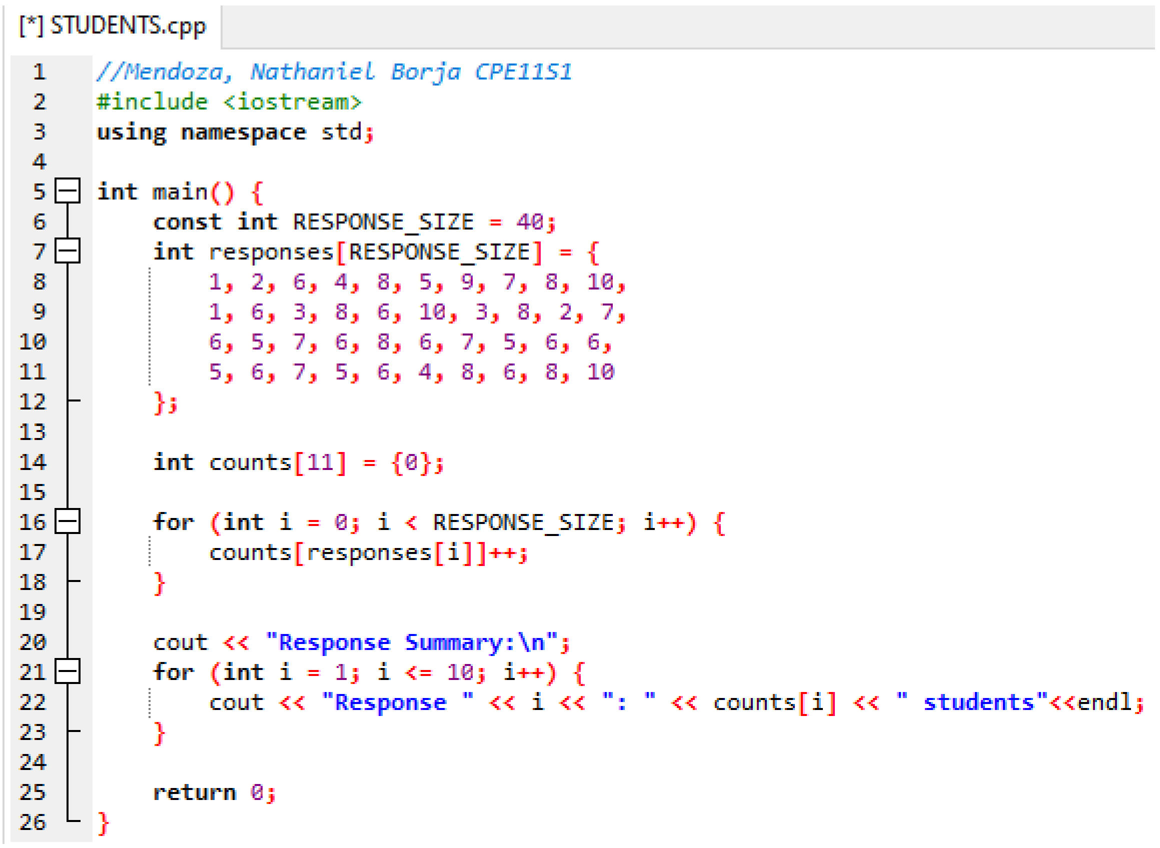
- The code concluded a #include <iostream> or Input/Output stream which gives access to code cout that's used for this code.
- using namespace std; has also been utilized for the code to shorten the coding.
- The “int main () {}” is the main function where the program starts running.
- The given set of array has a given size of 10 values
- The expected output consist of “*”, a Histogram so it will be implemented as char and given a variable that equals it.
- The expected output consists of an Element, Value, Histogram. Therefore, it will be printed as the header of the output.
- As for the outer loop of “for(int i = 0; i<10; i + +)” it will go through each element of the array, meaning 0 to 9 as the counting starts to 0. Next the “cout << i << " " << value[i] << " ";” is the one going to be printing the index or the element of an array and its value.
- Then the inner loop “for (int x = 0; x < value[i]; x++)” it will print “*” repeatedly depending on the value stored in the value[i]. Then after that it will print y as it was the defined character variable of *
- Then it will end after i and x is < 10.

Code Output:



```
Element       Value     Histrogram
  0            19     *******************
  1            3     ***
  2            15     ***************
  3            7     *******
  4            11     ***********
  5            9     *********
  6            13     *************
  7            5     *****
  8            17     *****************
  9            1     *

--------------------------------
Process exited after 0.01734 seconds with return value 0
Press any key to continue . . .
```

2. Code



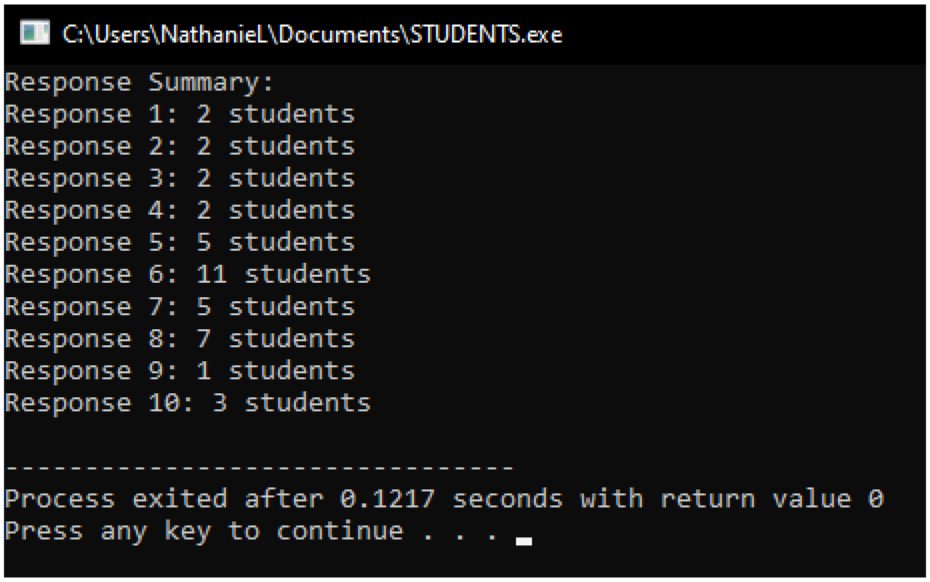
[*] STUDENTS.cpp

```cpp
//Mendoza, Nathaniel Borja CPE11S1
#include <iostream>
using namespace std;

int main() {
    const int RESPONSE_SIZE = 40;
    int responses[RESPONSE_SIZE] = {
        1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
        1, 6, 3, 8, 6, 10, 3, 8, 2, 7,
        6, 5, 7, 6, 8, 6, 7, 5, 6, 6,
        5, 6, 7, 5, 6, 4, 8, 6, 8, 10
    };

    int counts[11] = {0};

    for (int i = 0; i < RESPONSE_SIZE; i++) {
        counts[responses[i]]++;
    }

    cout << "Response Summary:\n";
    for (int i = 1; i <= 10; i++) {
        cout << "Response " << i << ": " << counts[i] << " students"<<endl;
    }

    return 0;
}
```

Code Explanation:

- The code concluded a #include <iostream> or Input/Output stream which gives access to code cout that's used for this code.
- using namespace std; has also been utilized for the code to shorten the coding.
- The “int main () {}” is the main function where the program starts running.
- The usage of const of “const int RESPONSE_SIZE = 40;” means the value inside it cannot be changed and which tells the compiler of an array will hold 40 responses.
- For the “int response[RESPONSE_SIZE]” it consists of 40 integers which will represent the student's response from 1 to 10.
- Then I used another array coded as “int counts[11] = {0};” so that it would count from 1 to 10. Why not counts[11]? Because, it would count from 0 to 9 only as c++ starts from 0. Meaning all values are set to 0 at the start. Therefore it was set as 11 so that it would count from 1-10 and 0 is ignored as there were not any students that did a 0 response.
- The for loop of “for(int i = 0; i < RESPONSE_SIZE; i++) {
counts[response[i]]++; } means that the response[i] gets the student answer and then it increments or adds 1 (+1) for the next loop. Then after that it will store how many students picked each response from the array.
- cout << "Response Summary:\n"; will just print or it will become the header before it produces the results.
- The code of “for (int i = 1; i <= 10; i++) { cout << "Response " << i << ": " << counts[i] << " students\n"; }” prints the summary as “i” goes from 1 to 10 (Till 10 are the possible responses)
- return 0; } ends the program.

Code Output:



```
Response Summary:
Response 1: 2 students
Response 2: 2 students
Response 3: 2 students
Response 4: 2 students
Response 5: 5 students
Response 6: 11 students
Response 7: 5 students
Response 8: 7 students
Response 9: 1 students
Response 10: 3 students

--------------------------------
Process exited after 0.1217 seconds with return value 0
Press any key to continue . . .
```

8. Conclusion

This activity has challenged me on developing my programming skills on creating and handling arrays. As these will be used and useful for data structures that would be storing multiple values of its same types making it easier to organize the same desired elements and process the data gathered. By working on 2 different problems and I learned how to declare arrays, initialize it and even print a non variable or a special character type such as “*” . Through the different exercises I can say that I can still improve my skills through printing arrays, getting its sum as the complicated formula is still making me confused and challenged as it's also been used on creating and sorting a bubble sort from what I have seen. Additionally, I learned about generating histograms by getting its values and making them generate the desired amount from the respected array element/number and counting surveys based on student responses from their ratings that have been gathered from 1 to 10. Overall, the activity shows that arrays are powerful and essential tools in programming. They allow us to store and manipulate sets of related data efficiently. By combining arrays with loops and conditional logic, in conclusion, I learned how to apply arrays in practical tasks like surveys and statistics. This activity strengthened my understanding of how arrays work in memory and their importance as a foundation for advanced programming.