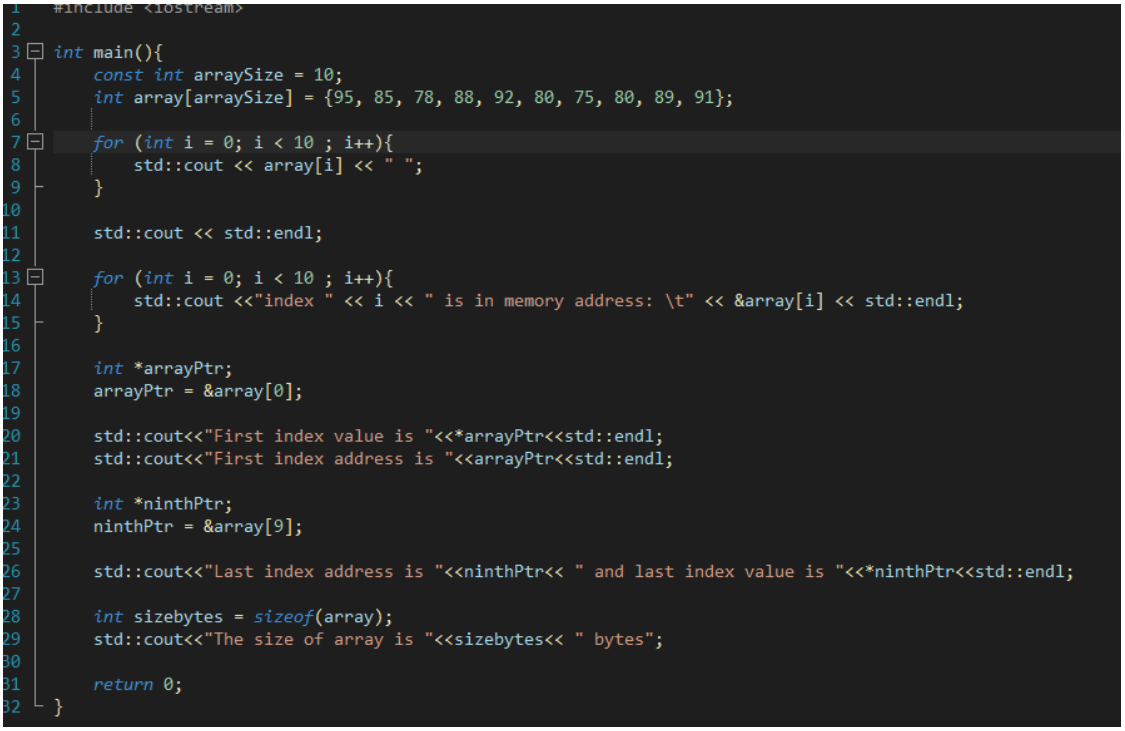
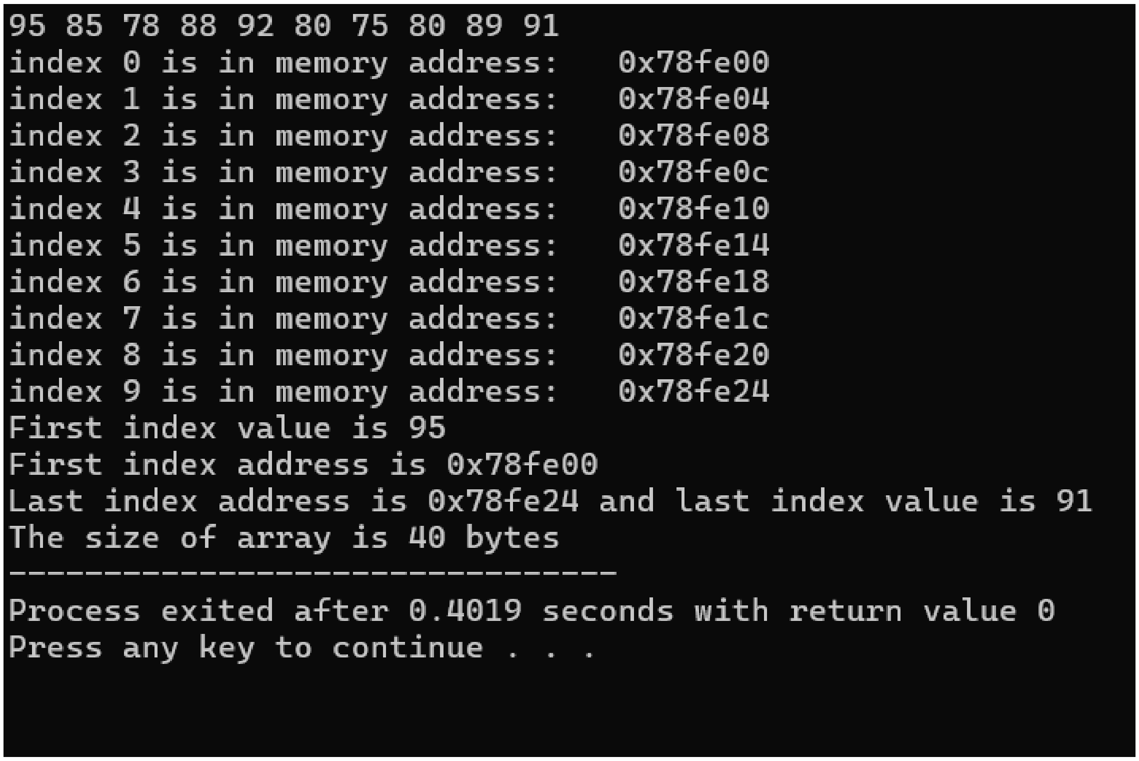
Seatwork 4.2	
Pointers	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:9/18/25**
Section:CPE11S1	**Date Submitted:9/18/25**
Name(s): Niel Vincent B. Condino	**Instructor: Engr. Jimlord M. Quejado**
6. Output	



```
#include <iostream>

int main(){
    const int arraySize = 10;
    int array[arraySize] = {95, 85, 78, 88, 92, 80, 75, 80, 89, 91};

    for (int i = 0; i < 10 ; i++){
        std::cout << array[i] << " ";
    }

    std::cout << std::endl;

    for (int i = 0; i < 10 ; i++){
        std::cout <<"index " << i << " is in memory address: \t" << &array[i] << std::endl;
    }

    int *arrayPtr;
    arrayPtr = &array[0];

    std::cout<<"First index value is "<<*arrayPtr<<std::endl;
    std::cout<<"First index address is "<<arrayPtr<<std::endl;

    int *ninthPtr;
    ninthPtr = &array[9];

    std::cout<<"Last index address is "<<ninthPtr<< " and last index value is "<<*ninthPtr<<std::endl;

    int sizebytes = sizeof(array);
    std::cout<<"The size of array is "<<sizebytes<< " bytes";

    return 0;
}
```



```
95 85 78 88 92 80 75 80 89 91
index 0 is in memory address:   0x78fe00
index 1 is in memory address:   0x78fe04
index 2 is in memory address:   0x78fe08
index 3 is in memory address:   0x78fe0c
index 4 is in memory address:   0x78fe10
index 5 is in memory address:   0x78fe14
index 6 is in memory address:   0x78fe18
index 7 is in memory address:   0x78fe1c
index 8 is in memory address:   0x78fe20
index 9 is in memory address:   0x78fe24
First index value is 95
First index address is 0x78fe00
Last index address is 0x78fe24 and last index value is 91
The size of array is 40 bytes
--------------------------------
Process exited after 0.4019 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

The first line in the function initializes a constant integer named arraySize with the value of 10. Then it creates an integer array with the name of “array” with a size of the value of the arraySize with the values :{95, 85, 78, 88, 92, 80, 75, 80, 89, 91}. The first for loop prints the values of the arrays. Below that there is a line break. The next for loop prints the index and their corresponding addresses in a format of "index (i) is in memory address: (address of that index)”. The next line initializes a pointer named arrayPtr and it points to the address of the first index in the array named “array”. The next lines print the pointers value and the address it points to using the format of "First index value is (value that the pointer points to)” and "First index address is (address the pointer points to). The next line also initializes a pointer named ninthPtr and it points to the address of the last index in the array named “array”. The next lines print the pointer's value and the address it points to using the format of "Last index address is (address the pointer points to) and last index value is (value that the pointer points to). Then it initializes an int variable named sizebytes with the value of the function sizeof() used on the array named “array”. After that it prints the size of the array that is occupied in the memory using the format "The size of array is (value of sizebytes) bytes". Lastly it returns 0.

8. Conclusion

Today, I learned the concept of pointers. They are a special type of variable with the memory address as its own value. It can be used to get the address of a certain element in memory and also the value of it. I also learned about the & operator. I learned how it can be used in order to get a certain memory address.The procedure of the output part of the activity is not that hard as it is guided, although I changed some variable names. It just showed how pointers are used in c++. I think this showed the basics on how to use this kind of variable.The supplementary is also just an analysis of the code in the output section. It is quite easy to do but takes quite a while to explain the full parts of it.Overall, the lessons are still pretty understandable. Pointer’s functions are pretty interesting to learn. However, there are still things to learn from this topic.

9. Assessment Rubric