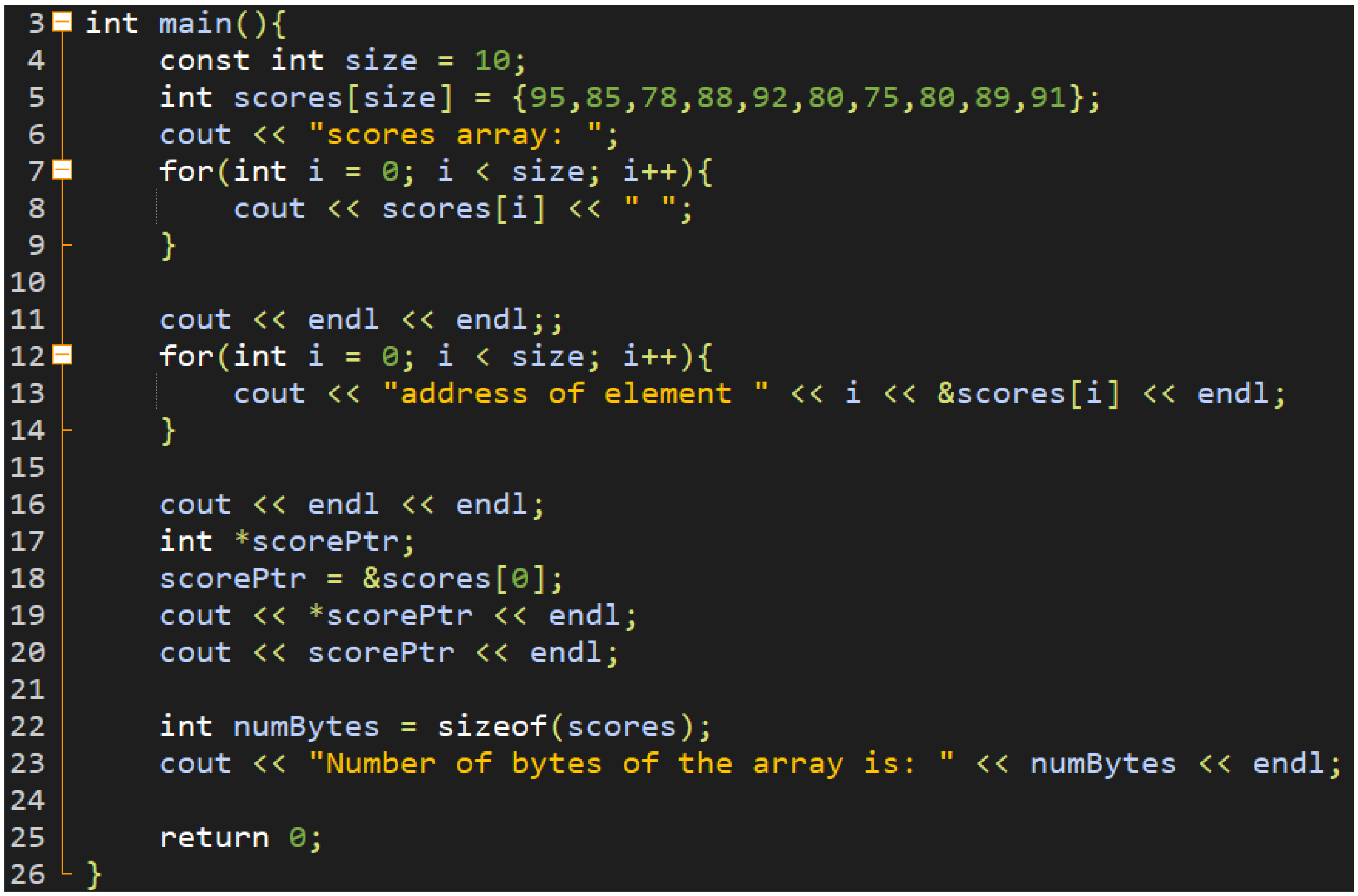
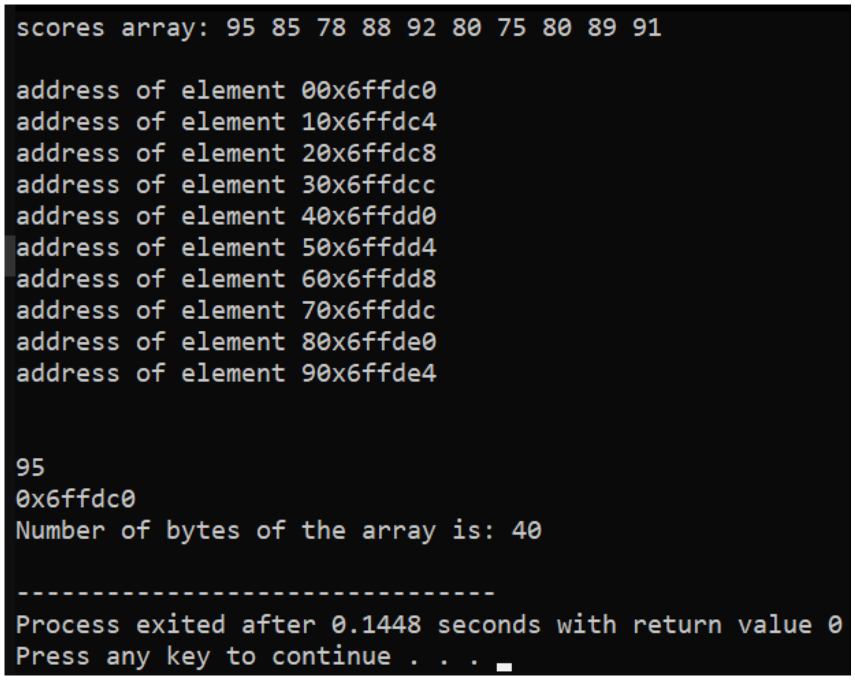
Seatwork 4.2	
Pointers	
Course Code: CPE 007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 9/18/2025**
Section: CPE11S1	**Date Submitted: 9/18/2025**
Name(s): Juan Paulo C. Lara	**Instructor: Engr. Jimlord M. Quejado**
6. Output	



```
 3  int main(){
 4      const int size = 10;
 5      int scores[size] = {95,85,78,88,92,80,75,80,89,91};
 6      cout << "scores array: ";
 7      for(int i = 0; i < size; i++){
 8          cout << scores[i] << " ";
 9      }
10
11      cout << endl << endl;;
12      for(int i = 0; i < size; i++){
13          cout << "address of element " << i << &scores[i] << endl;
14      }
15
16      cout << endl << endl;
17      int *scorePtr;
18      scorePtr = &scores[0];
19      cout << *scorePtr << endl;
20      cout << scorePtr << endl;
21
22      int numBytes = sizeof(scores);
23      cout << "Number of bytes of the array is: " << numBytes << endl;
24
25      return 0;
26  }
```



```
scores array: 95 85 78 88 92 80 75 80 89 91

address of element 00x6ffdc0
address of element 10x6ffdc4
address of element 20x6ffdc8
address of element 30x6ffdcc
address of element 40x6ffdd0
address of element 50x6ffdd4
address of element 60x6ffdd8
address of element 70x6ffddc
address of element 80x6ffde0
address of element 90x6ffde4


95
0x6ffdc0
Number of bytes of the array is: 40

--------------------------------
Process exited after 0.1448 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

During execution, it begins at line 4 initializing a constant integer variable size to 10. The declared size is then used for an array of 10 elements allocated by scores: 95,85,78,88,92,80,75,80,89,91. Line 6 outputs a header message: “scores array: “. Next line, a for loop outputs the 10 elements with spaces after each number. After the spacing, another for loop at line 12 outputs the memory address of the elements. As you can see at the first three output memory addresses, there are numbers 0, 4, and 8. These show the way arrays are stored in memory by the computer through the 4 bytes allocated by int. Moving over, line 17 initializes the pointer to the integers. The next line tells the pointer to look for array element 0 or the first element. *scorePtr tells to dereference the pointer to print the stored data in that chosen element. The next line without the * symbol outputs the memory address of the chosen element. Finally, Line 22 outputs the memory allocated by the array, which is 40 bytes through multiplying 4 bytes (from int) by 10 integers (from array).

8. Conclusion

Ending this activity, I gained understanding in the use of pointers through its symbols * and &, and the reading of memory addresses and its allocation depending on the type of declaration used like int, float, double, long double, etc. Pointers allowed me to see the part of a stored memory that a computer can understand, which are hexadecimals like 0xffffff. Reading the code, two operators are used: *asterisk and &. *The asterisk symbol before a variable points to a type of initialization such as int or char. It also dereferences by getting the address to get a certain value. The and symbol stores the memory address to the variable and also acts as a reference of another variable. Overall, I gained more understanding in how computer memory works through code, how it stores values and allocates them in memory, and how code can interface with memory to find the actual content of the inputted information.