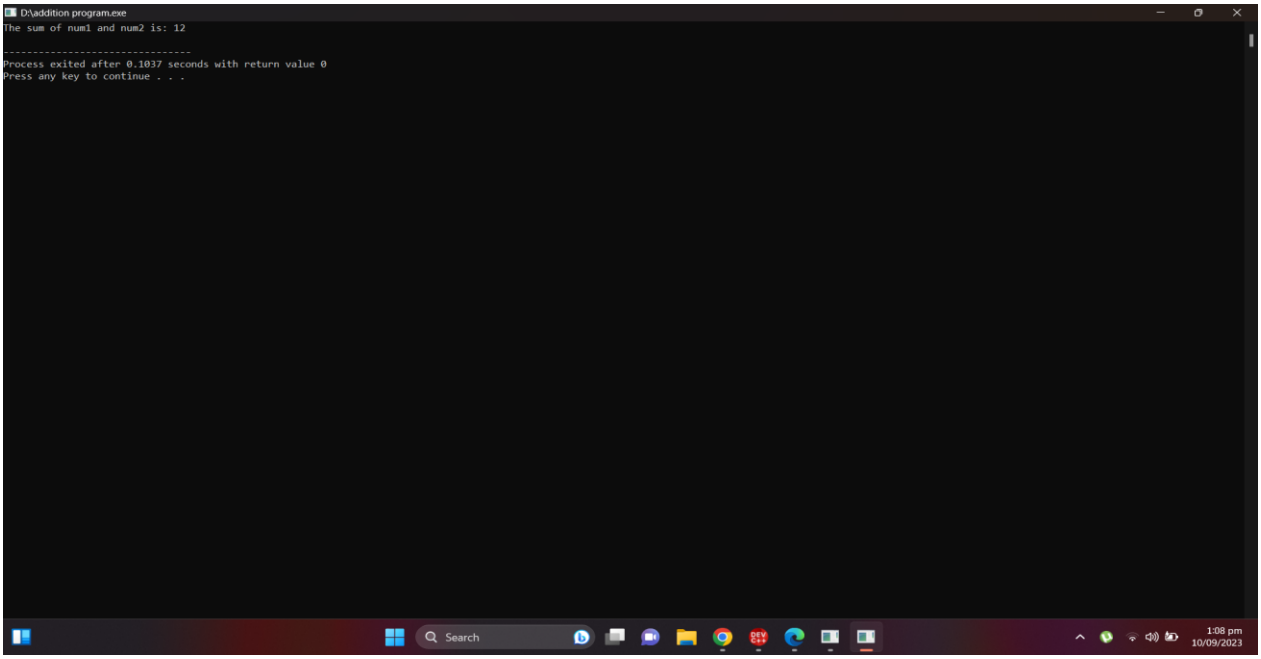
ASSIGNMENT # 1

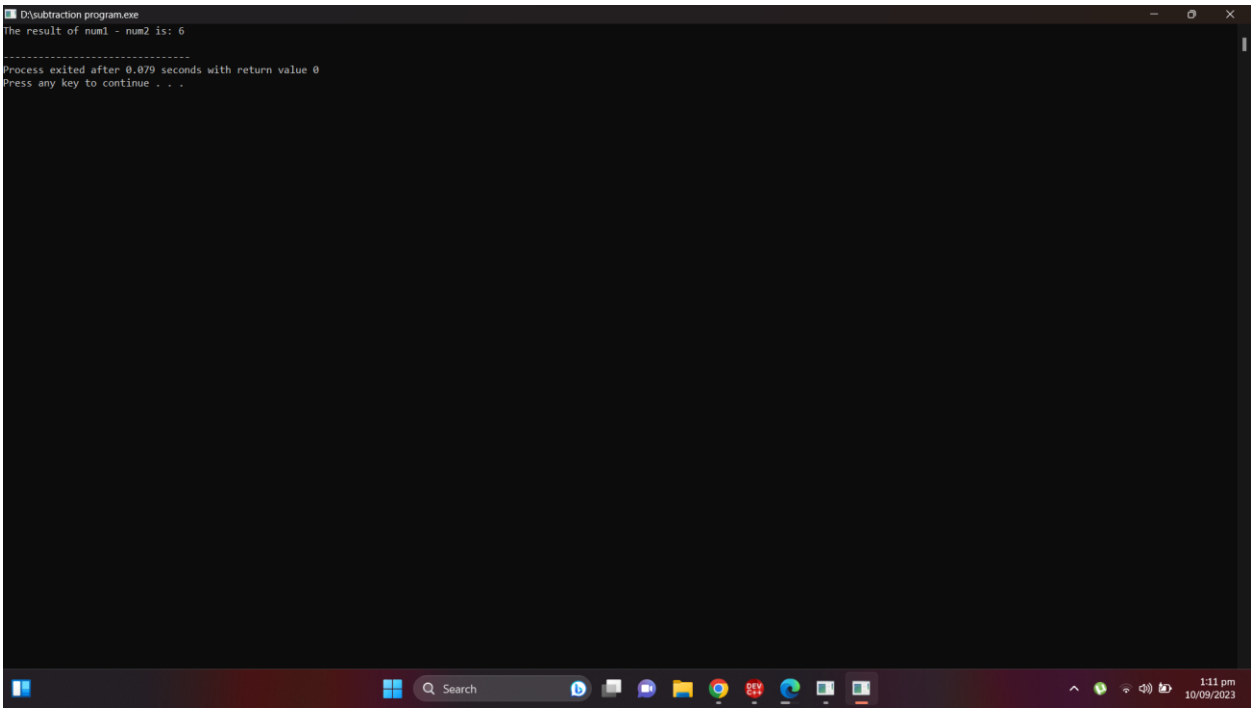
Program# 1: Addition Program

```
#include <iostream>

using namespace std;

int main() {

    int num1 = 5;

    int num2 = 7;

    int result;

   int* ptr1 = &num1;

   int* ptr2 = &num2;

  result = (*ptr1) + (*ptr2);

    cout << "The sum of num1 and num2 is: " << result <<endl;

    return 0;

}
```



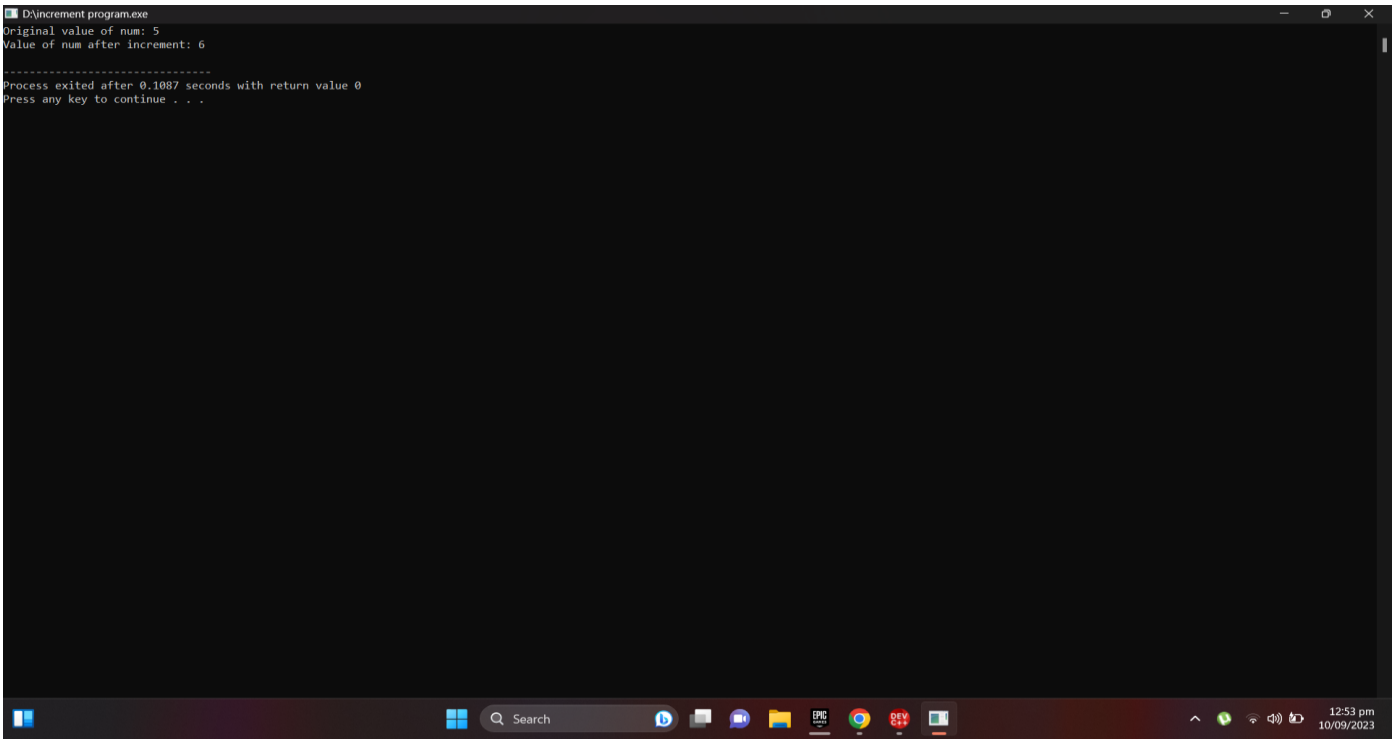
Program# 2: Subtraction Program

```
#include <iostream>

using namespace std;

int main () {
   int num1 = 10;
   int num2 = 4;
   int result;
   int* ptr1 = &num1;
   int* ptr2 = &num2;
       result = (*ptr1) - (*ptr2);
   cout << "The result of num1 - num2 is: " << result <<endl;
   return 0;
}
```



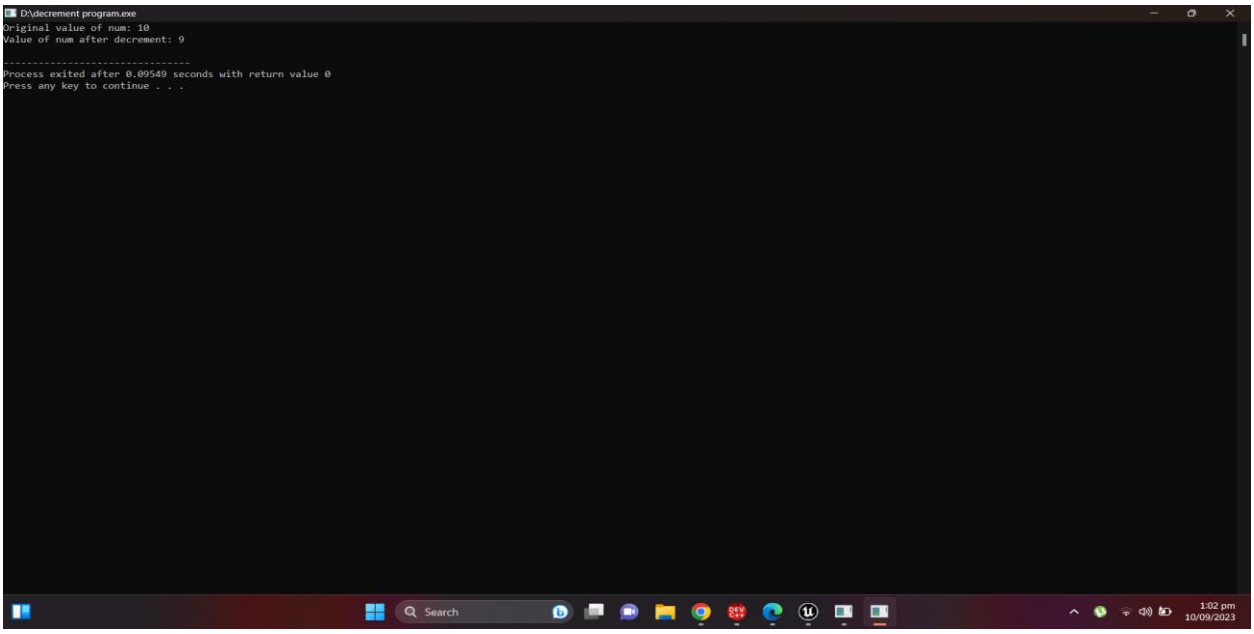
Program# 3: Increment Program

```
#include <iostream>

using namespace std;

int main () {
   int num = 5;
   int* ptr = &num;
   cout << "Original value of num: " << num <<endl;
   (*ptr)++;
   cout << "Value of num after increment: " << num <<endl;
    return 0;
}
```



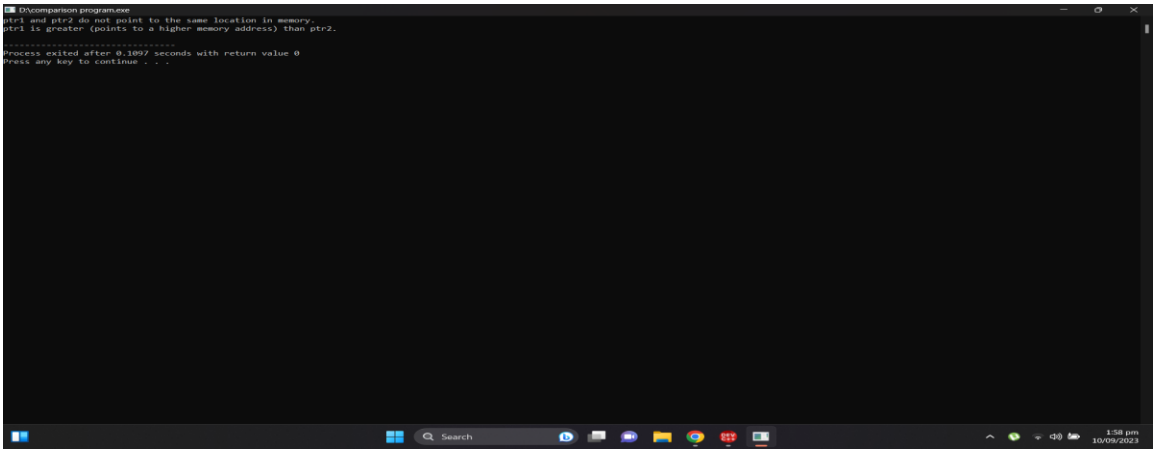
Program# 4: Decrement Program

```
#include <iostream>

using namespace std;

int main () {
   int num = 10;
   int* ptr = &num;
   cout << "Original value of num: " << num <<endl;
   (*ptr)--;
   cout << "Value of num after decrement: " << num <<endl;
   return 0;
}
```



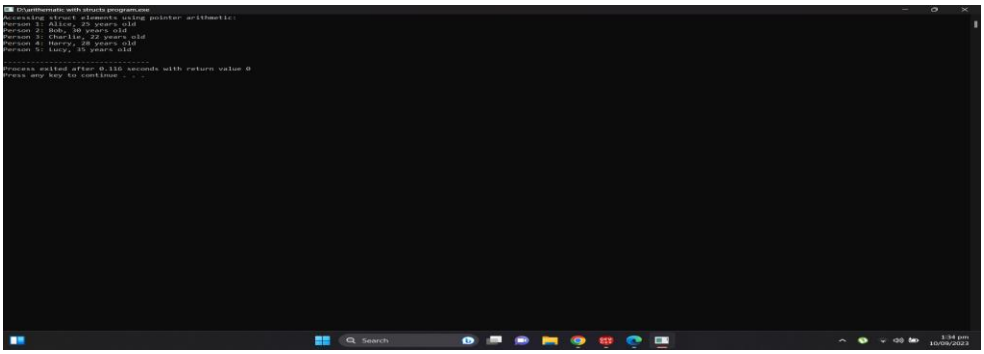
Program# 5: Comparison Program

```
#include <iostream>

using namespace std;

int main() {
    int num1 = 10;
    int num2 = 20;
    int* ptr1 = &num1;
    int* ptr2 = &num2;
   if (ptr1 == ptr2) {
        cout << "ptr1 and ptr2 point to the same location in memory." <<endl;
    } else {
        cout << "ptr1 and ptr2 do not point to the same location in memory." <<endl;
    }
    if (ptr1 > ptr2) {
        cout << "ptr1 is greater (points to a higher memory address) than ptr2." <<endl;
    } else if (ptr1 < ptr2) {
    cout << "ptr1 is smaller (points to a lower memory address) than ptr2." <<endl;
    } else {
        cout << "ptr1 and ptr2 have the same memory address." <<endl;
    }
   return 0;
}
```



Program# 6: Arithmetic with structs program Program

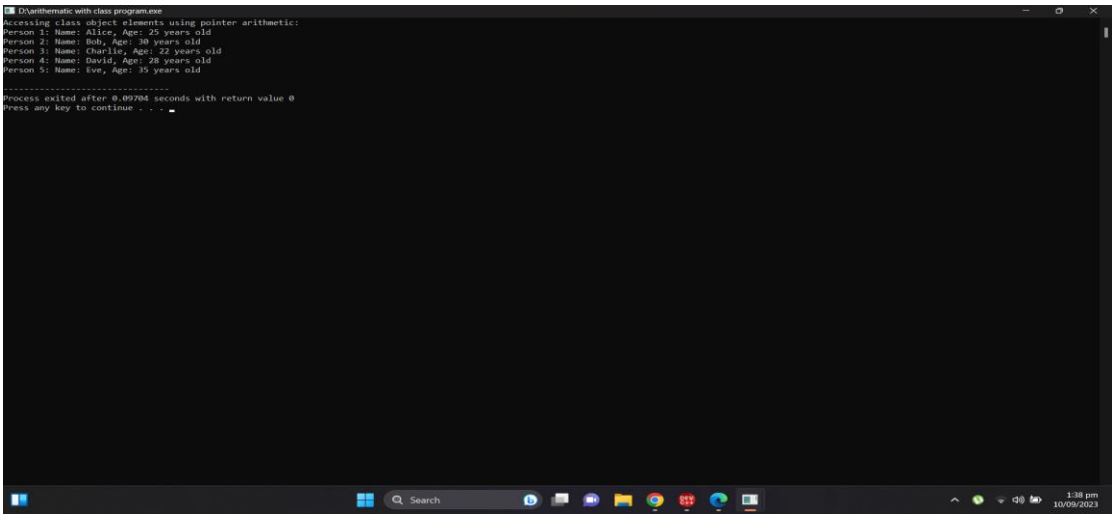
```
#include <iostream>
#include <string>
using namespace std;
struct Person {
   string name;
   int age;
    };
    int main () {
   Person people [] = {
      {"Alice", 25},
      {"Bob", 30},
      {"Charlie", 22},
      {"Harry", 28},
      {"Lucy", 35}
   };
   Person* ptr = people;
   cout << "Accessing struct elements using pointer arithmetic:" <<endl;
   for (int i = 0; i < 5; i++)
      {
cout << "Person " << i + 1 << ": " << (ptr + i)->name << ", " << (ptr + i)->age << " years old" <<endl;
   }
  return 0;
}
```



Program# 7: Arithmetic with Class program

```
#include <iostream>
#include <string>
using namespace std;
class Person {
public:
   Person (const string& name, int age) : name(name), age(age) {}
    void printInfo () {
      cout << "Name: " << name << ", Age: " << age << " years old" <<endl;}
private:
   string name;
   int age;
};
int main () {
   Person people [] = {
      {"Alice", 25},
      {"Bob", 30},
      {"Charlie", 22},
      {"David", 28},
      {"Eve", 35}
   };
    Person* ptr = people;
```

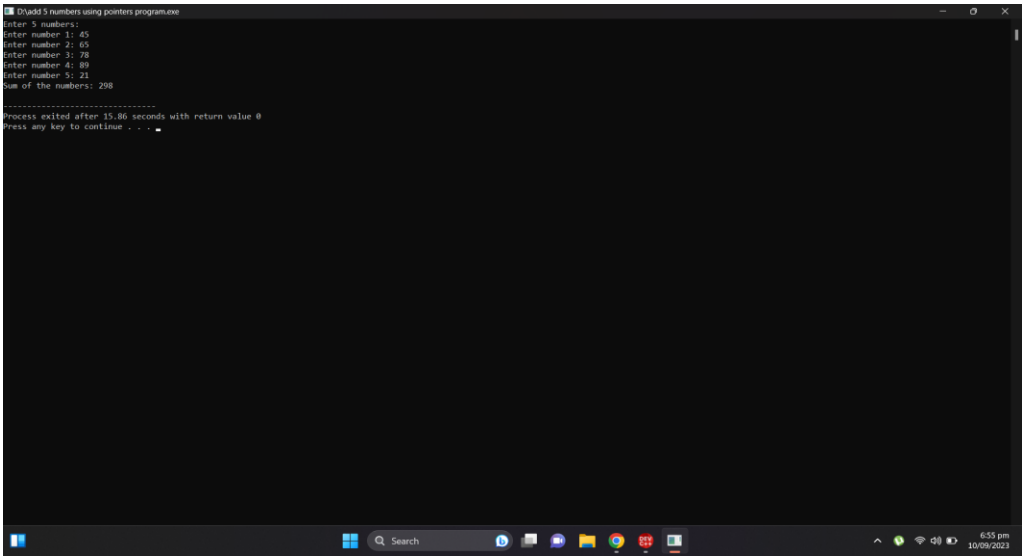
```
    cout<< "Accessing class object elements using pointer arithmetic:" <<endl;
    for (int i = 0; i < 5; i++) {
        cout << "Person " << i + 1 << ": ";
        (ptr + i)->printInfo();
    }
    return 0;
}
```



Program# 8: Adding 5 numbers using pointers program

```
#include <iostream>
using namespace std;
int main () {
    int numbers [5];
    cout << "Enter 5 numbers:" <<endl;
    for (int i = 0; i < 5; ++i) {
        cout<< "Enter number " << i + 1 << ": ";
        cin >> numbers[i];
    }
    int* ptr = numbers;
    int sum = 0;
    for (int i = 0; i < 5; ++i) {
```

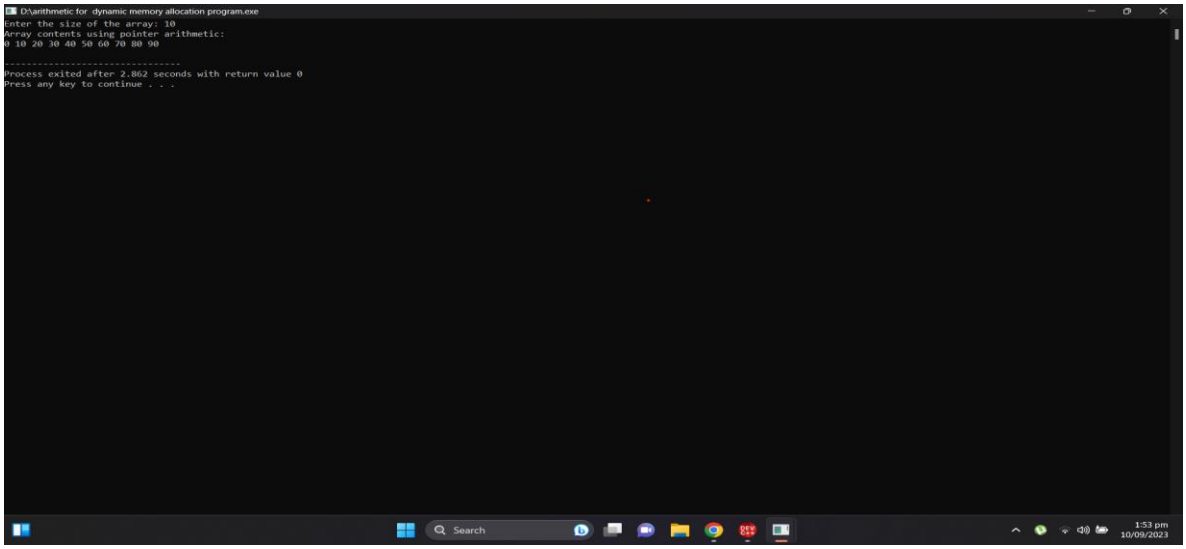
```
        sum += *ptr;
        ptr++;
    }
   cout << "Sum of the numbers: " << sum <<endl;
   return 0;
}
```



Program# 9: Arithmetic for dynamic memory allocation program

```
#include <iostream>
using namespace std;
int main() {
    int size;
    cout << "Enter the size of the array: ";
    cin >> size;
    int* dynamicArray = new int[size];
    for (int i = 0; i < size; i++) {
        dynamicArray[i] = i * 10;
    }
    cout << "Array contents using pointer arithmetic:" <<endl;
    for (int i = 0; i < size; i++) {
```

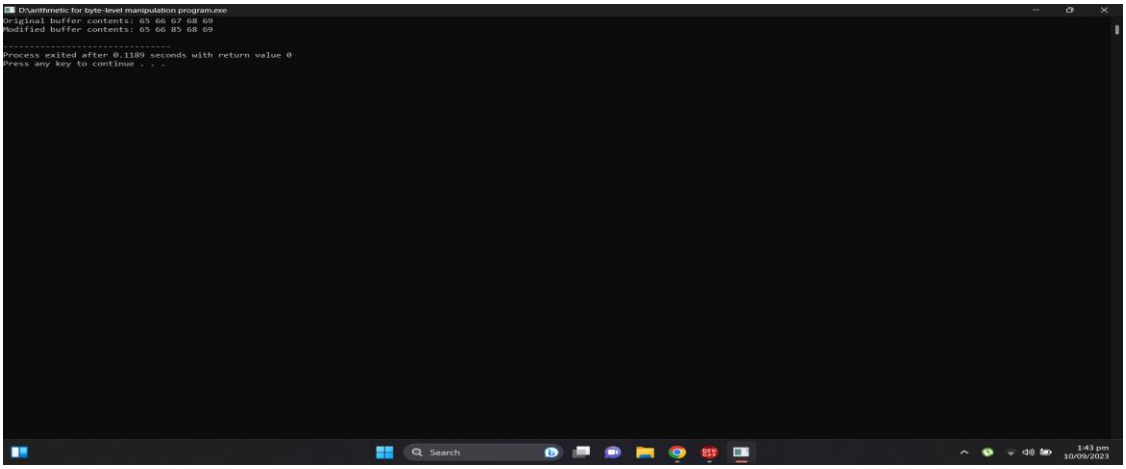
```
        cout << *(dynamicArray + i) << " ";
    }
    cout <<endl;
    delete[] dynamicArray;
    return 0;
}
```



Program# 10: Arithmetic for byte-level manipulation program

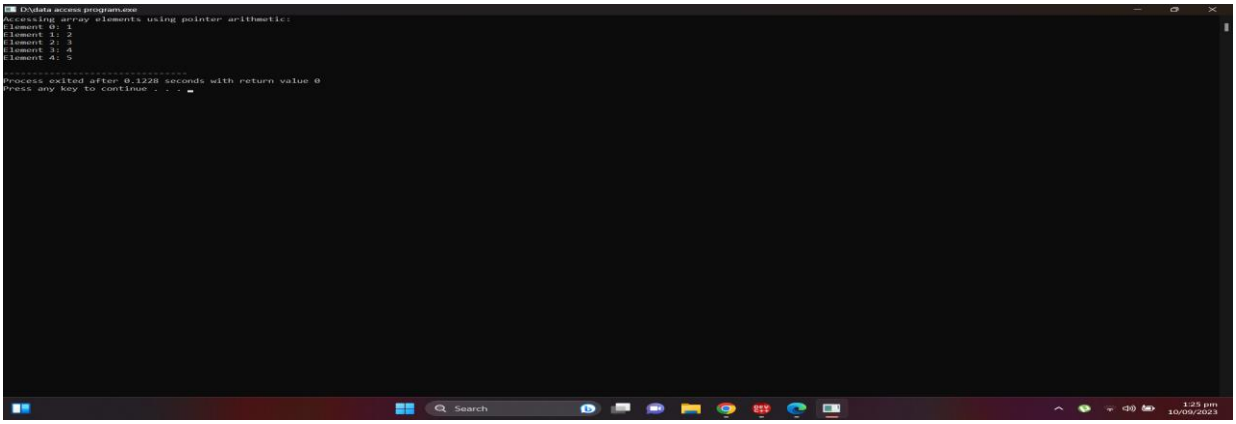
```
#include <iostream>
using namespace std;
int main() {
    char buffer[5] = {0x41, 0x42, 0x43, 0x44, 0x45};
char* ptr = buffer;
    cout << "Original buffer contents: ";
    for (int i = 0; i < 5; i++) {
        cout << static_cast<int>(buffer[i]) << " ";
    }
    cout << endl;
*(ptr + 2) = 0x55;
    cout << "Modified buffer contents: ";
```

```
    for (int i = 0; i < 5; i++) {
        cout << static_cast<int>(buffer[i]) << " ";
    }
    cout << endl;
    return 0;
}
```



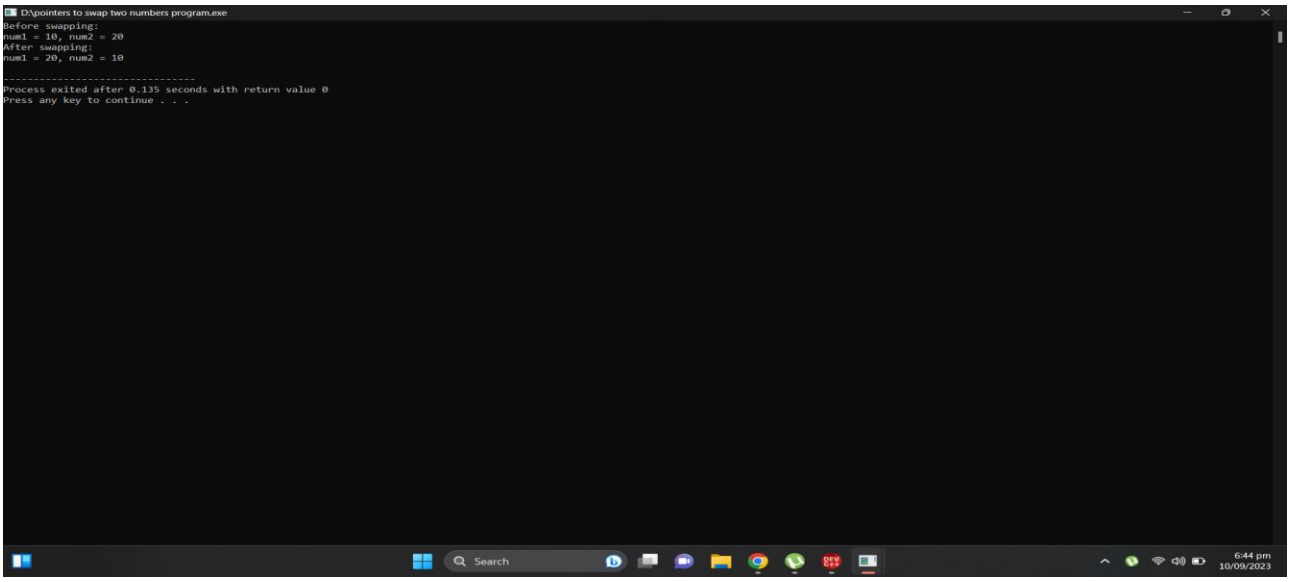
Program# 11: Data Access Program

```
#include <iostream>
using namespace std;
int main() {
    int numbers[] = {1, 2, 3, 4, 5};
    int* ptr = numbers;
    cout<<"Accessing array elements using pointer arithmetic:" <<endl;
    for (int i = 0; i < 5; i++) {
    cout<< "Element " << i << ": " << *(ptr + i) <<endl;
    }
    return 0;
    }
```



Program# 12: Pointers to swap two numbers program

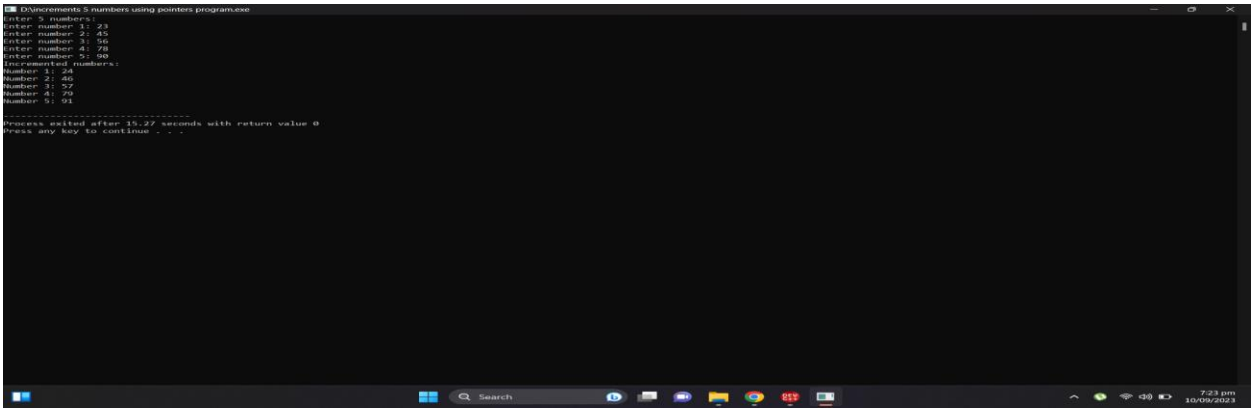
```
#include <iostream>

using namespace std;

void swapNumbers(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int num1 = 10;
    int num2 = 20;
    cout << "Before swapping:" <<endl;
    cout << "num1 = " << num1 << ", num2 = " << num2 <<endl;
    swapNumbers(&num1, &num2);
cout << "After swapping:" <<endl;
    cout << "num1 = " << num1 << ", num2 = " << num2 <<endl;
    return 0;
}
```



Program# 13: Increments 5 numbers using pointers program

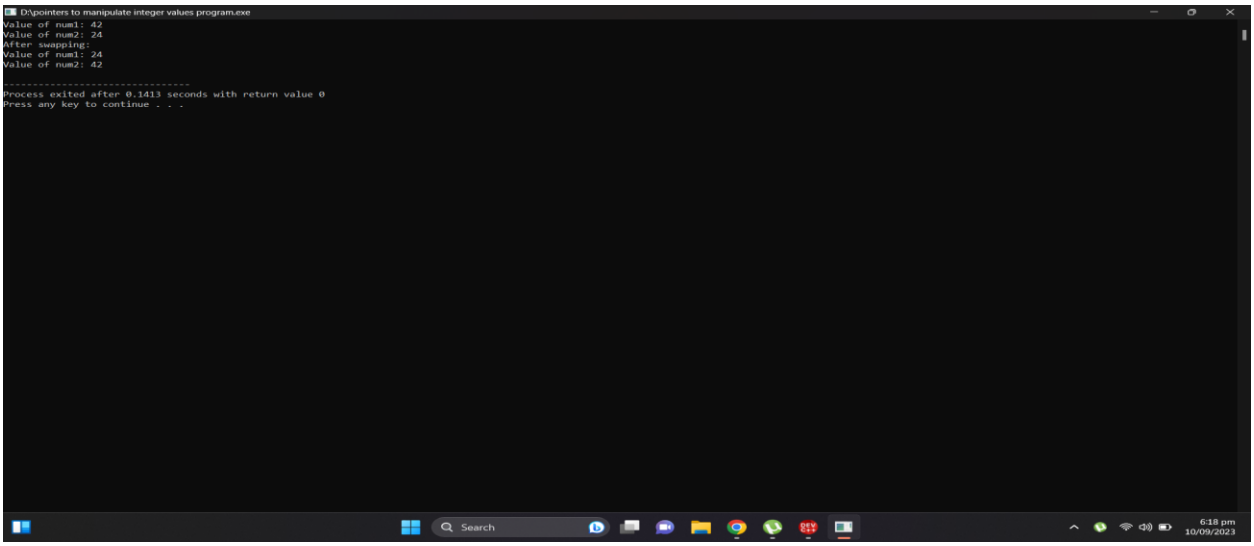
```
#include <iostream>

using namespace std;

int main() {
   int numbers[5];
   cout << "Enter 5 numbers:" <<endl;
   for (int i = 0; i < 5; ++i) {
      cout << "Enter number " << i + 1 << ": ";
      cin >> numbers[i];
   }
   int* ptr = numbers;
    for (int i = 0; i < 5; ++i) {
      (*ptr)++;
      ptr++;
   }
   cout << "Incremented numbers:" <<endl;
   for (int i = 0; i < 5; ++i) {
      cout << "Number " << i + 1 << ": " << numbers[i] <<endl;
   }
```

```
    return 0;
}
```



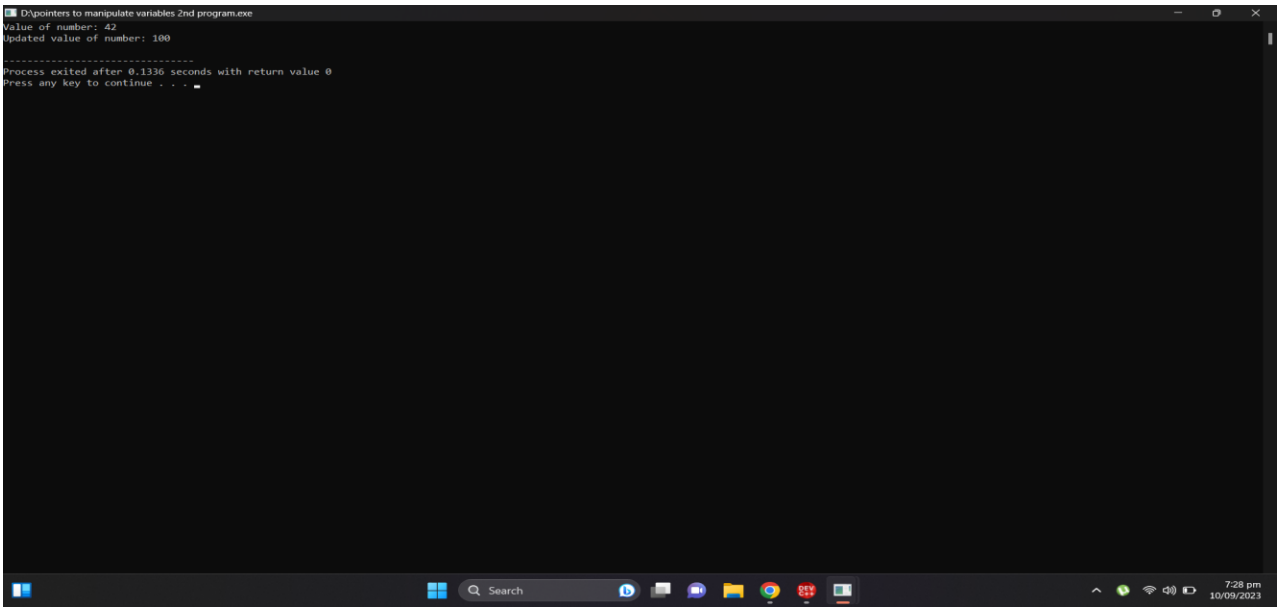
Program# 14: Pointers to manipulate integer values program

```
#include <iostream>
using namespace std;
int main() {
    int num1 = 42;
    int num2 = 24;
    int* ptr1 = &num1;
    int* ptr2 = &num2;
cout << "Value of num1: " << *ptr1 <<endl;
    cout << "Value of num2: " << *ptr2 <<endl;
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
cout << "After swapping:" <<endl;
    cout << "Value of num1: " << *ptr1 <<endl;
    cout << "Value of num2: " << *ptr2 <<endl;
     return 0;
}
```



Program# 15: Pointers to manipulate variables 2nd program

```
#include <iostream>

using namespace std;

int main() {
   int number = 42;
   int* pointer = &number;
   cout << "Value of number: " << *pointer <<endl;
   *pointer = 100;
   cout << "Updated value of number: " << number <<endl;
   return 0;
}
```


D:\pointers to manipulate variables 2nd program.exe
Value of number: 42
Updated value of number: 100

Process exited after 0.1336 seconds with return value 0
Press any key to continue . . .
Search
7:28 pm
10/09/2023