|  |  |
| --- | --- |
| download | COMSATS University Islamabad, Vehari Campus Department of Computer Science |

**Class: BCS-SP22-4B Submission Deadline: 10 Sep2023**

**Subject: Data Structures and Algorithms-Lab Instructor: Yasmeen Jana Max Marks: 10 Reg. No: SP22-BCS-087**

**Email:** [**yasmeenjana@cuivehari.edu.pk**](mailto:yasmeenjana@cuivehari.edu.pk)

**You can ask queries related to Lab Activities on the above email.**

# Activity 1: Creating a Github Account

Create a GitHub Account. Make a repository with the name “**DSA\_Lab”. Mention the link here after the account creation.**

**SOLUTION:**

**LINK:** <https://github.com/Mubashir-087/DSA_LAB>

# Activity 2: 15 Programs related to Pointers

**Program 01: Pointer Declaration and Initialization:**

#include <stdio.h>

int main() {

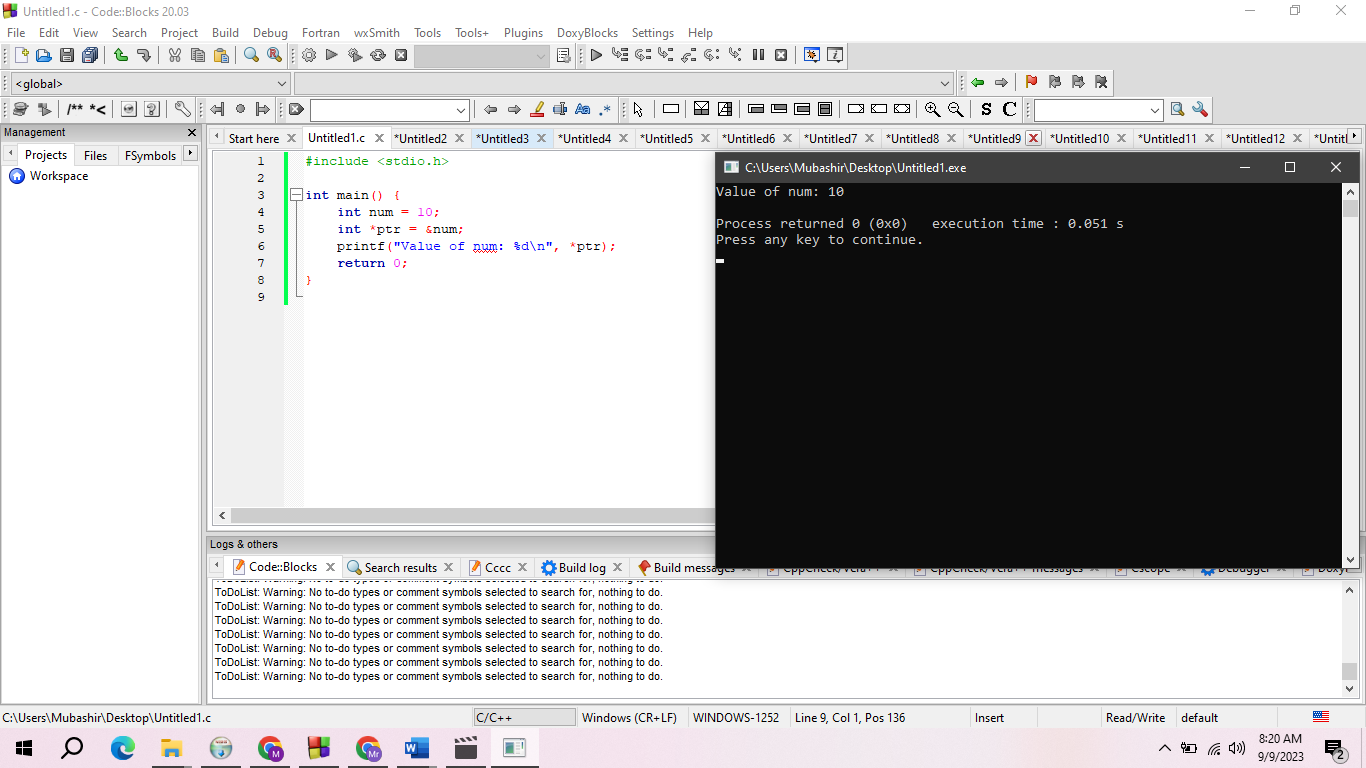
int num = 10;

int \*ptr = &num;

printf("Value of num: %d\n", \*ptr);

return 0;

}



**Program 02: Pointer Arithmetic:**

#include <stdio.h>

int main() {

int arr[] = {1, 2, 3, 4, 5};

int \*ptr = arr;

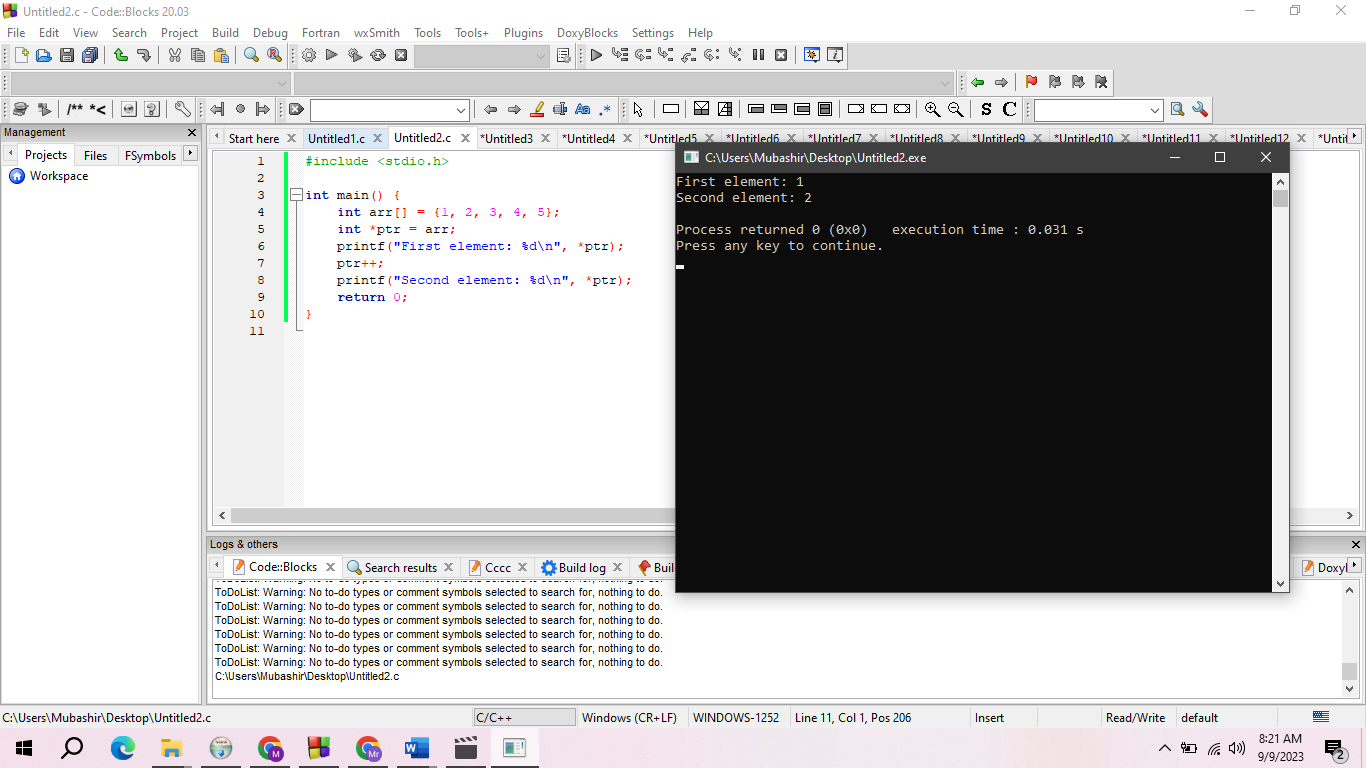
printf("First element: %d\n", \*ptr);

ptr++;

printf("Second element: %d\n", \*ptr);

return 0;

}



**Program 03: Pointer to Pointer:**

#include <stdio.h>

int main() {

int num = 10;

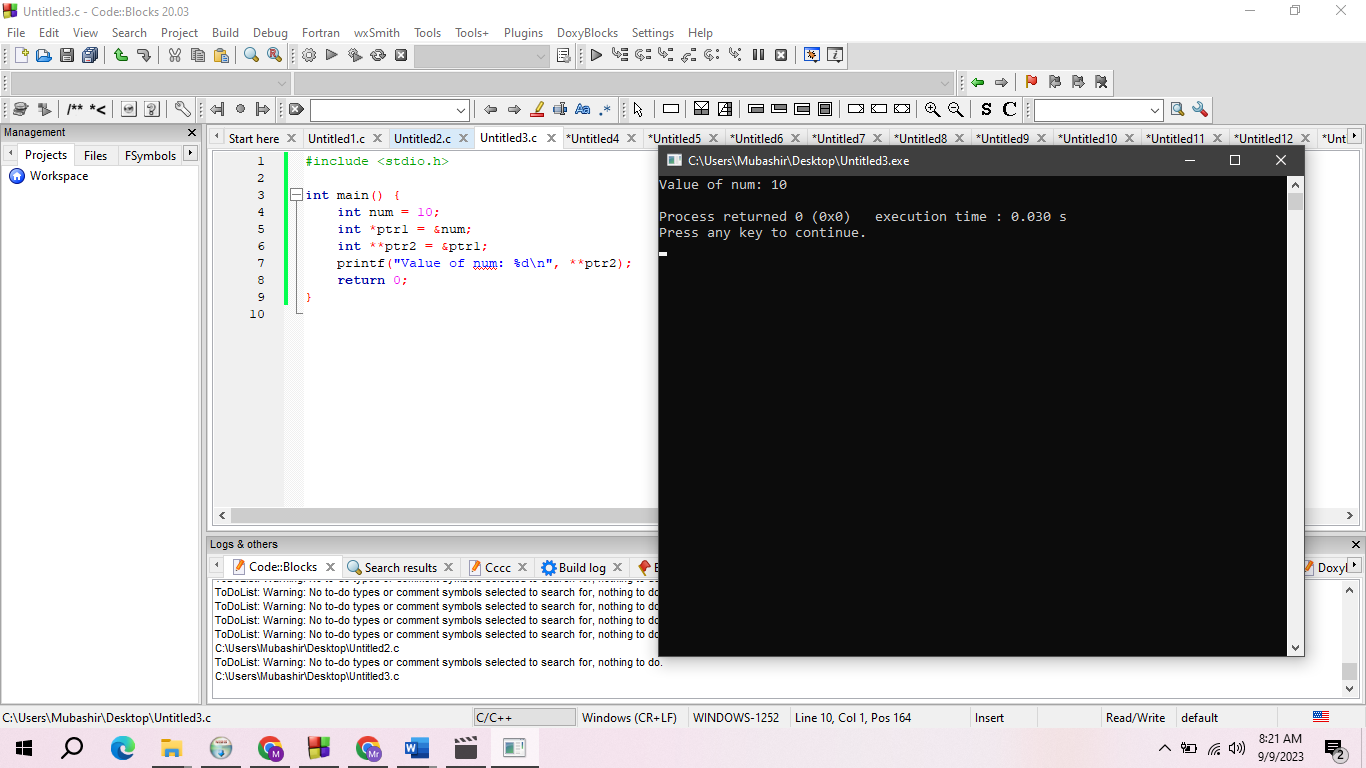
int \*ptr1 = &num;

int \*\*ptr2 = &ptr1;

printf("Value of num: %d\n", \*\*ptr2);

return 0;

}



**Program 04: Passing Pointers to Functions:**

#include <stdio.h>

void increment(int \*x) {

(\*x)++;

}

int main() {

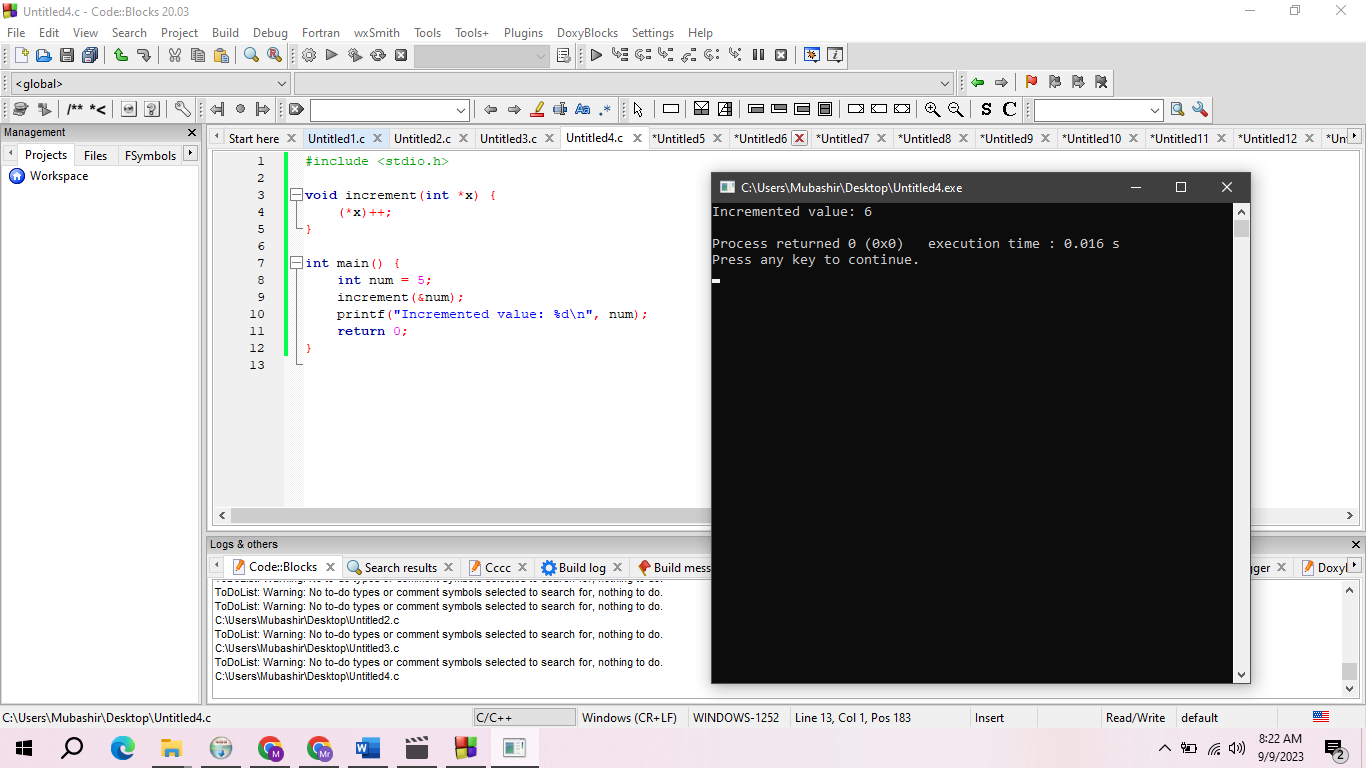
int num = 5;

increment(&num);

printf("Incremented value: %d\n", num);

return 0;

}



**Program 05: Arrays and Pointers:**

#include <stdio.h>

int main() {

int arr[] = {1, 2, 3, 4, 5};

int \*ptr = arr;

for (int i = 0; i < 5; i++) {

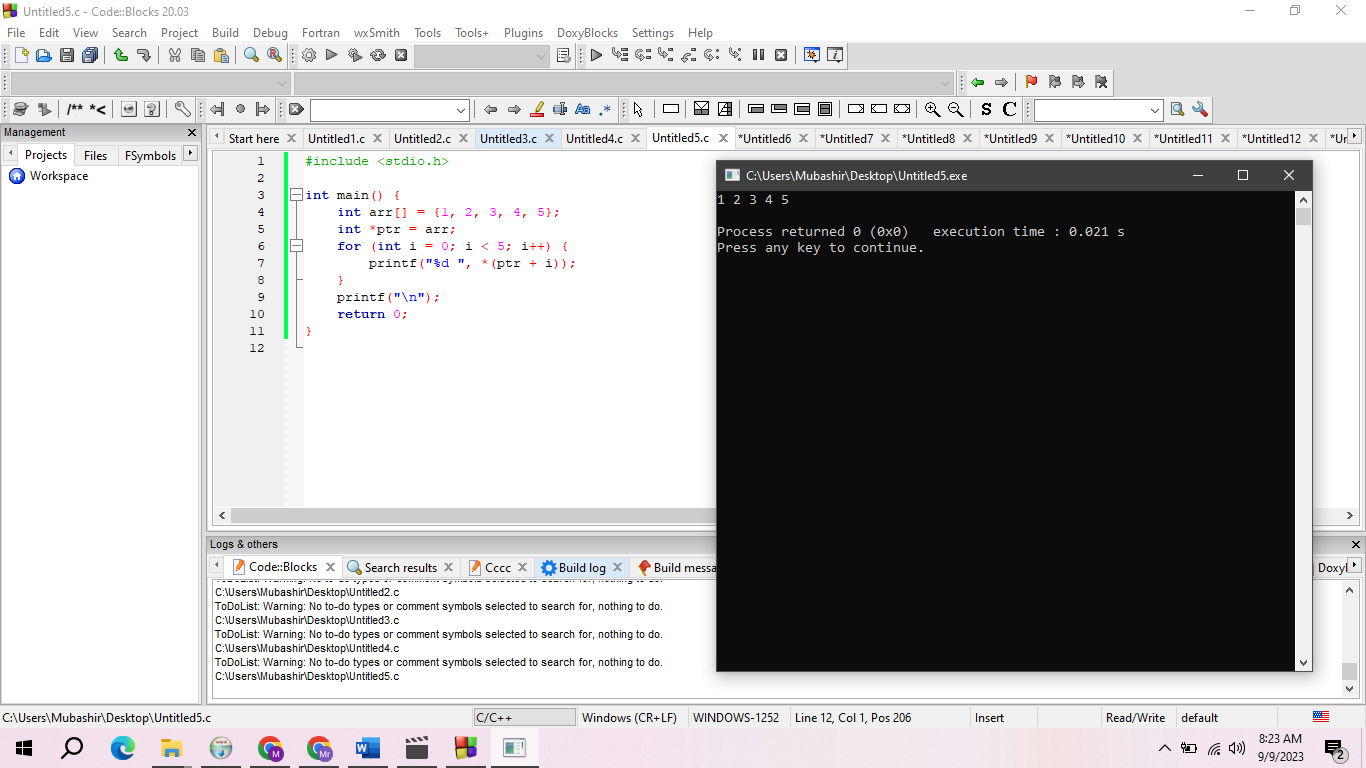
printf("%d ", \*(ptr + i));

}

printf("\n");

return 0;

}



**Program 06: Pointer and Strings:**

#include <stdio.h>

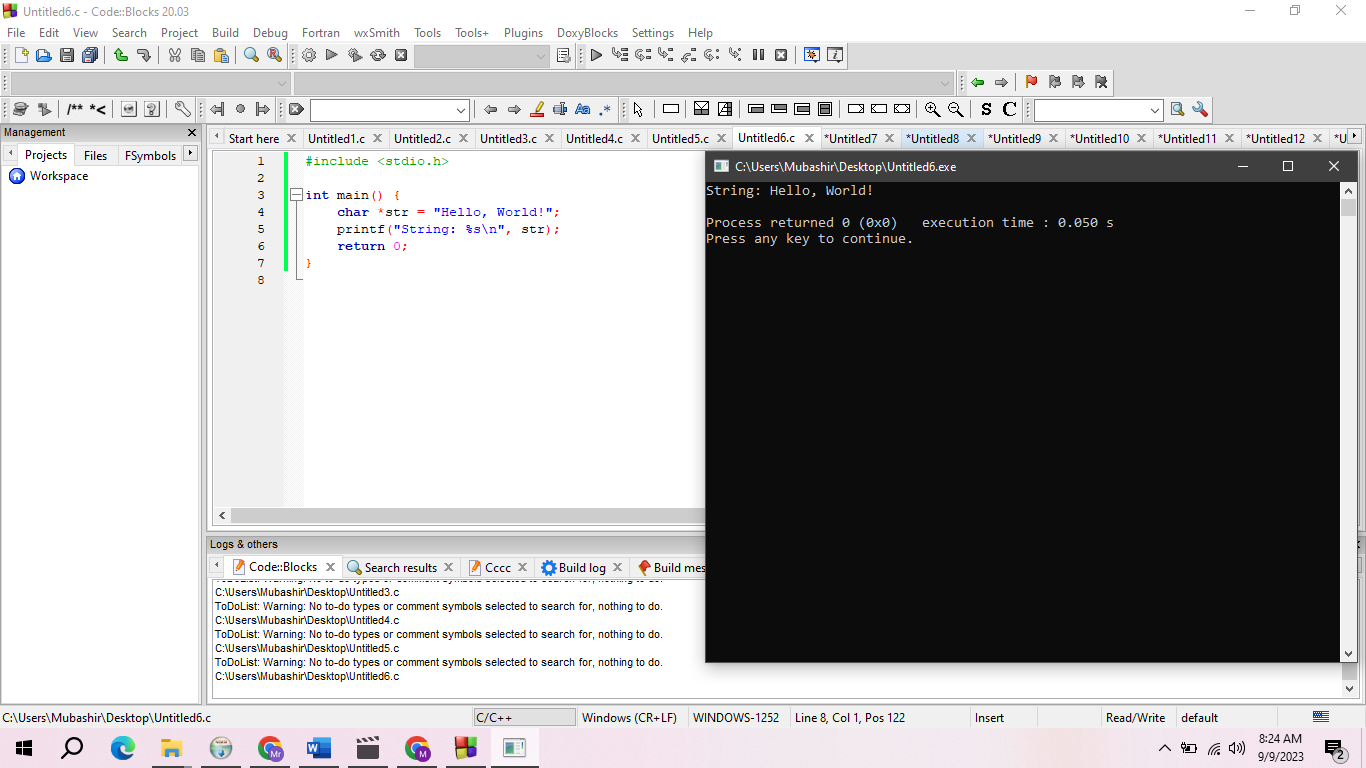
int main() {

char \*str = "Hello, World!";

printf("String: %s\n", str);

return 0;

}



**Program 07: Dynamic Memory Allocation (malloc and free):**

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*ptr = (int \*)malloc(sizeof(int));

if (ptr != NULL) {

\*ptr = 42;

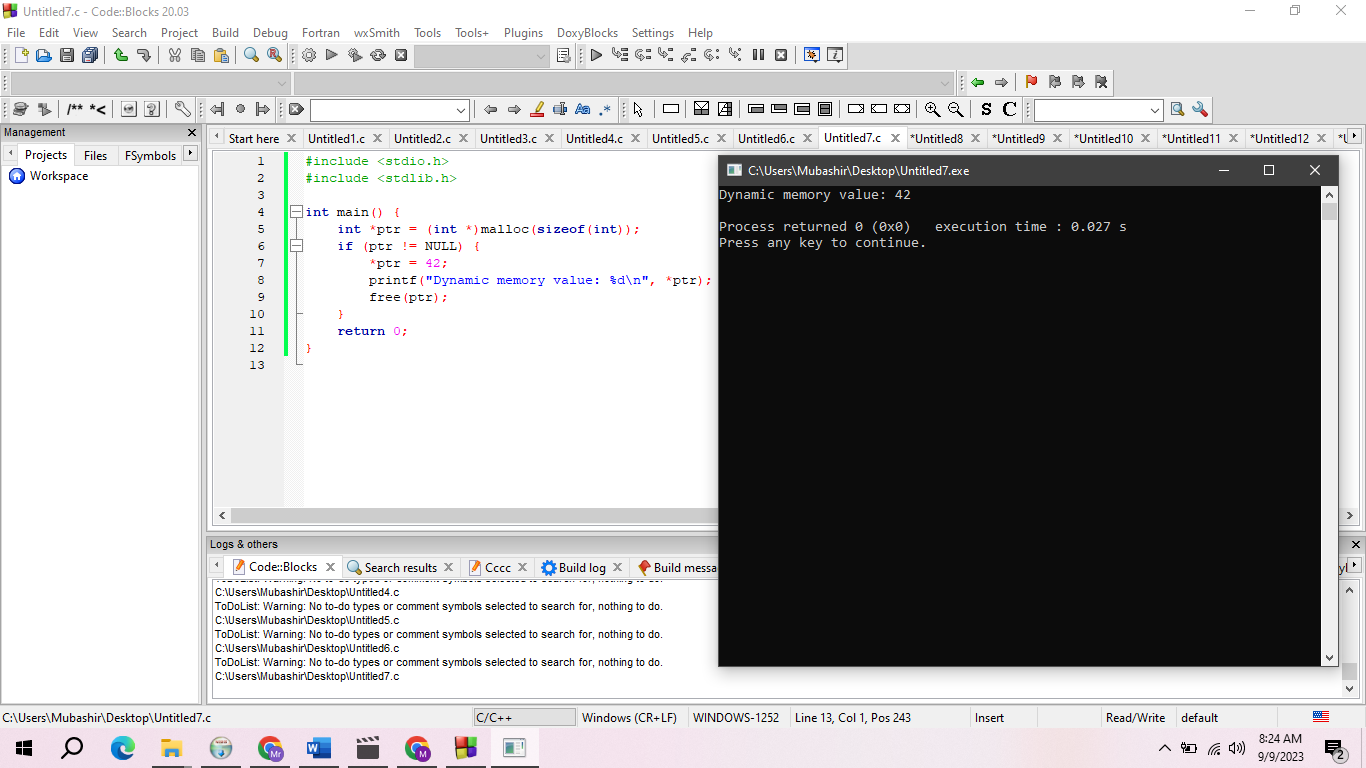
printf("Dynamic memory value: %d\n", \*ptr);

free(ptr);

}

return 0;

}



**Program 08: Pointer to Array:**

#include <stdio.h>

int main() {

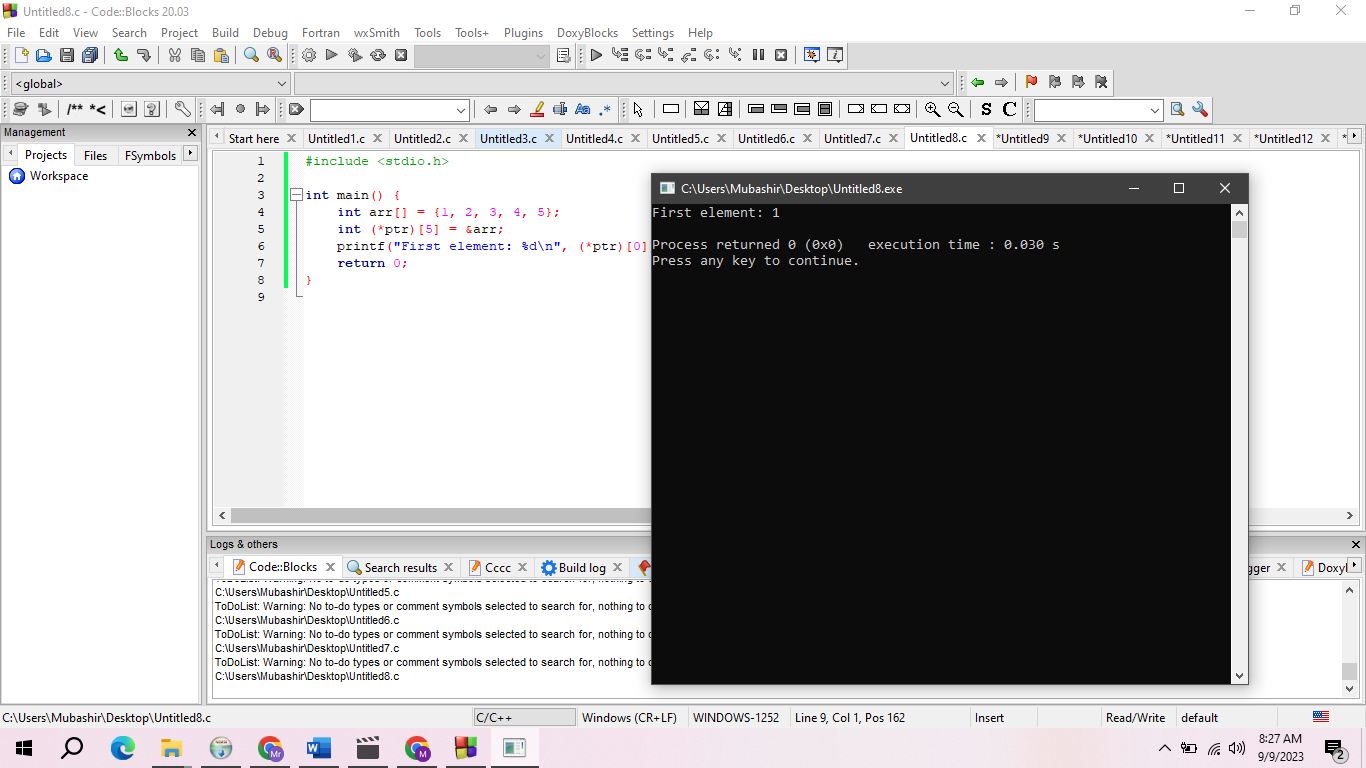
int arr[] = {1, 2, 3, 4, 5};

int (\*ptr)[5] = &arr;

printf("First element: %d\n", (\*ptr)[0]);

return 0;

}



**Program 09: Pointer to Function:**

#include <stdio.h>

int add(int a, int b) {

return a + b;

}

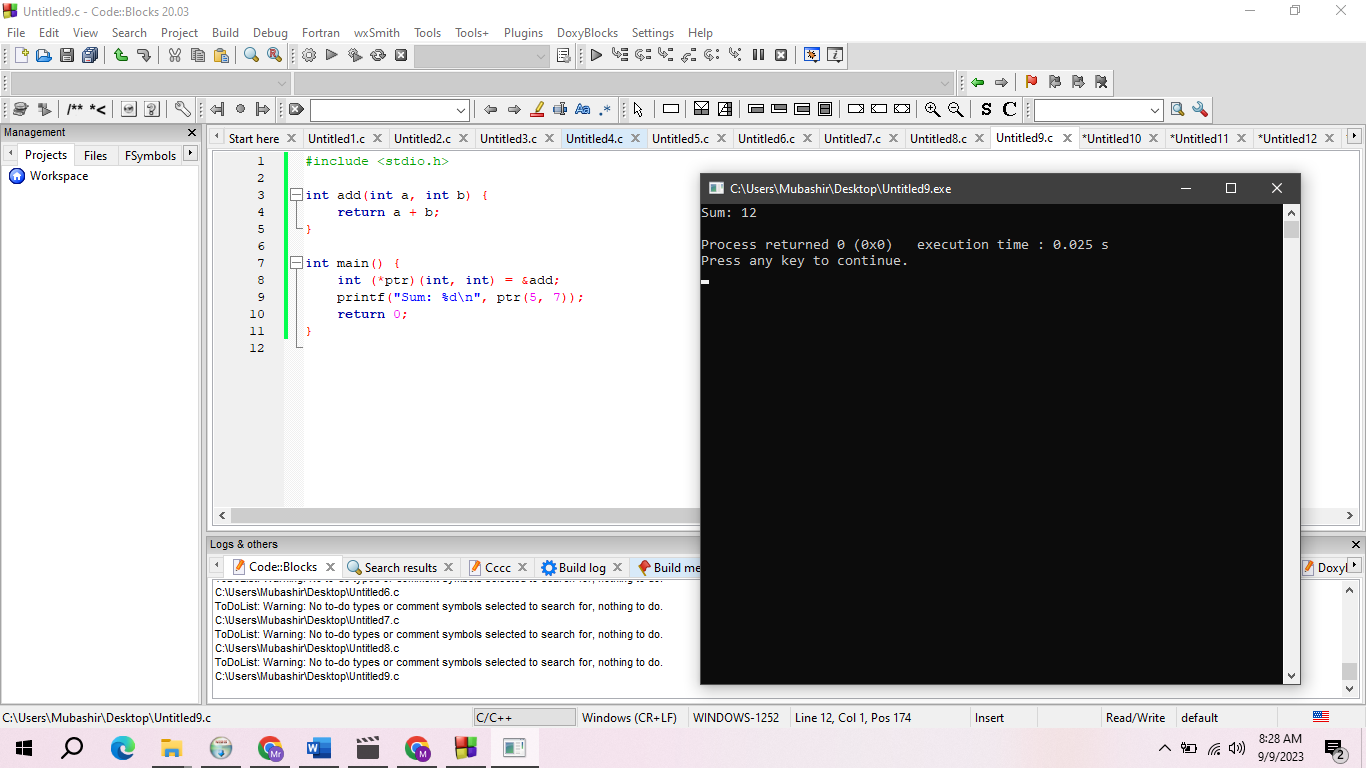
int main() {

int (\*ptr)(int, int) = &add;

printf("Sum: %d\n", ptr(5, 7));

return 0;

}



**Program 10: Pointer Comparison:**

#include <stdio.h>

int main() {

int num1 = 10, num2 = 20;

int \*ptr1 = &num1, \*ptr2 = &num2;

if (ptr1 == ptr2) {

printf("Pointers are equal.\n");

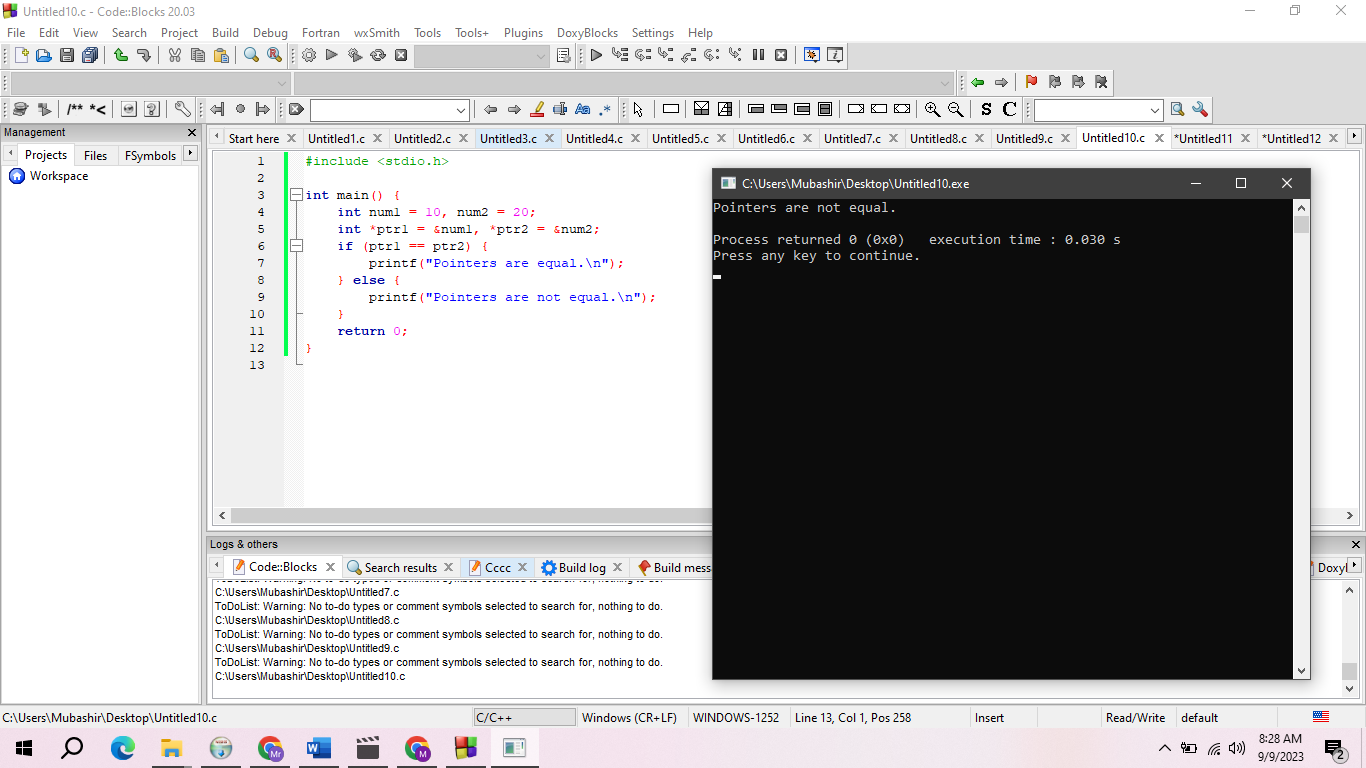
} else {

printf("Pointers are not equal.\n");

}

return 0;

}



**Program 11: Void Pointer (Generic Pointer):**

#include <stdio.h>

int main() {

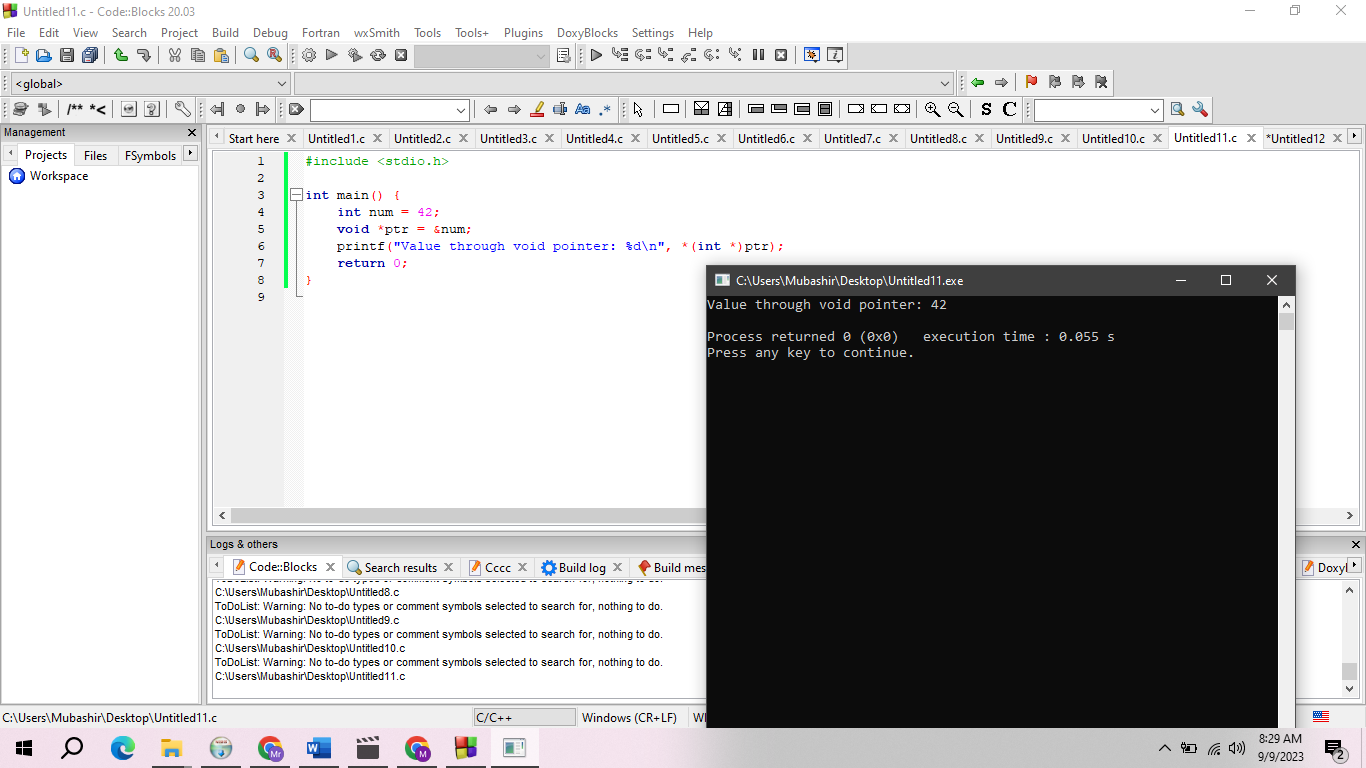
int num = 42;

void \*ptr = &num;

printf("Value through void pointer: %d\n", \*(int \*)ptr);

return 0;

}



**Program 12: Pointer to Structures:**

#include <stdio.h>

struct Point {

int x, y;

};

int main() {

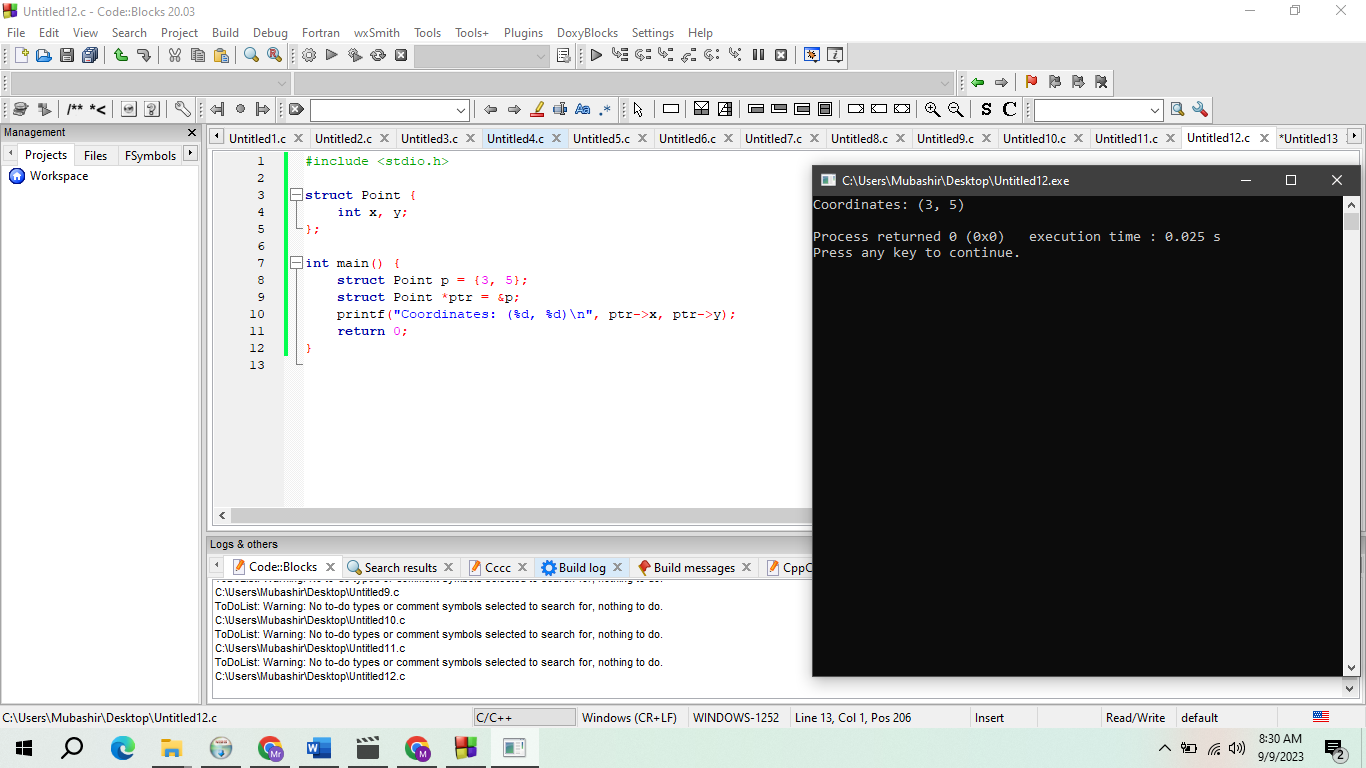
struct Point p = {3, 5};

struct Point \*ptr = &p;

printf("Coordinates: (%d, %d)\n", ptr->x, ptr->y);

return 0;

}



**Program 13:** **Pointer to Array of Structures:**

#include <stdio.h>

struct Student {

char name[50];

int age;

};

int main() {

struct Student students[3];

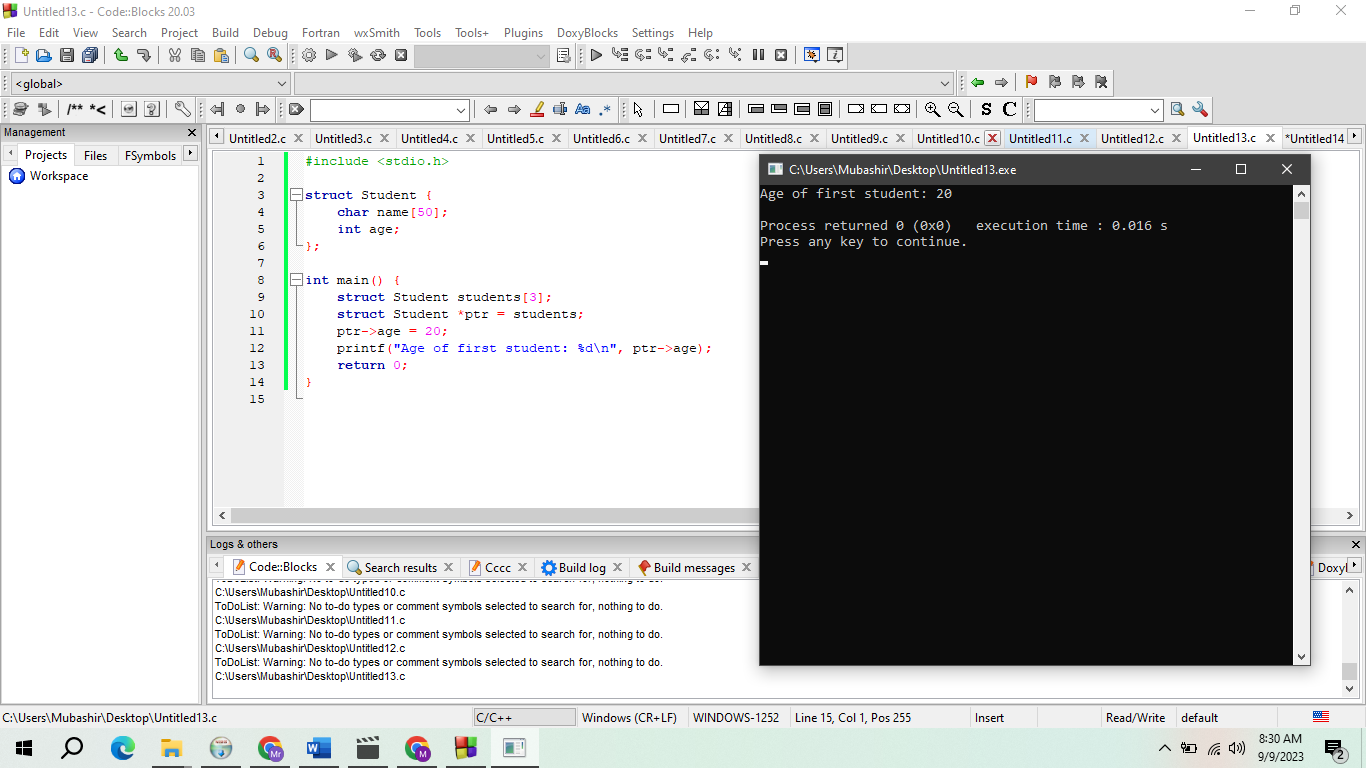
struct Student \*ptr = students;

ptr->age = 20;

printf("Age of first student: %d\n", ptr->age);

return 0;

}



**Program 14: Pointer to Function Pointer:**

#include <stdio.h>

int add(int a, int b) {

return a + b;

}

int subtract(int a, int b) {

return a - b;

}

int main() {

int (\*ptr1)(int, int) = &add;

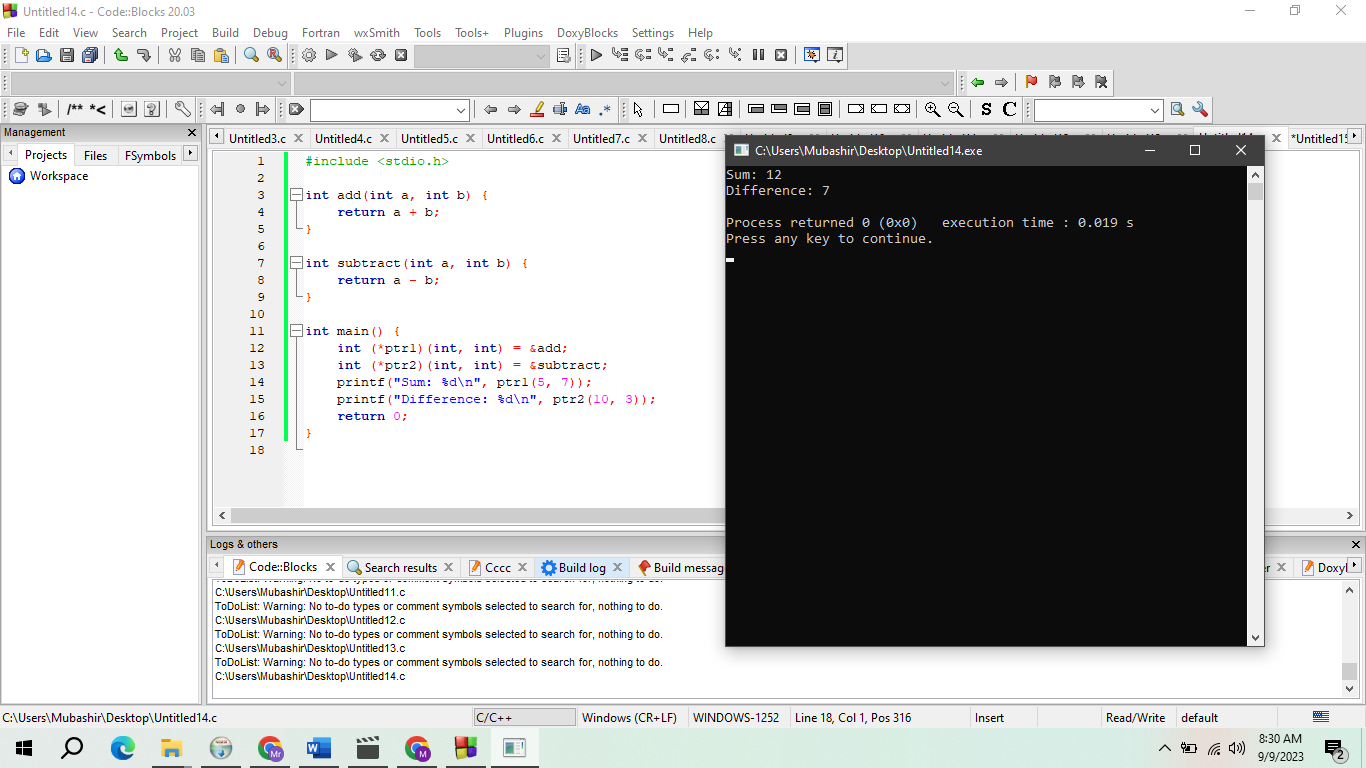
int (\*ptr2)(int, int) = &subtract;

printf("Sum: %d\n", ptr1(5, 7));

printf("Difference: %d\n", ptr2(10, 3));

return 0;

}



**Program 15: Pointer Arithmetic with Character Strings:**

#include <stdio.h>

int main() {

char str[] = "Hello";

char \*ptr = str;

while (\*ptr != '\0') {

printf("%c ", \*ptr);

ptr++;

}

printf("\n");

return 0;

}

