



COMSATS University Islamabad, Vehari Campus

Name: Uzair Mukhtar

Registration Number: SP22-BCS-081

Department: Computer Science

Course: DSA Lab

Instructor: Yasmeeen Jana

Assignment: 1st

Section: B

Program # 1

Output:

```
#include <stdio.h>
```

```
int main() {
```

```
    int num = 10;
```

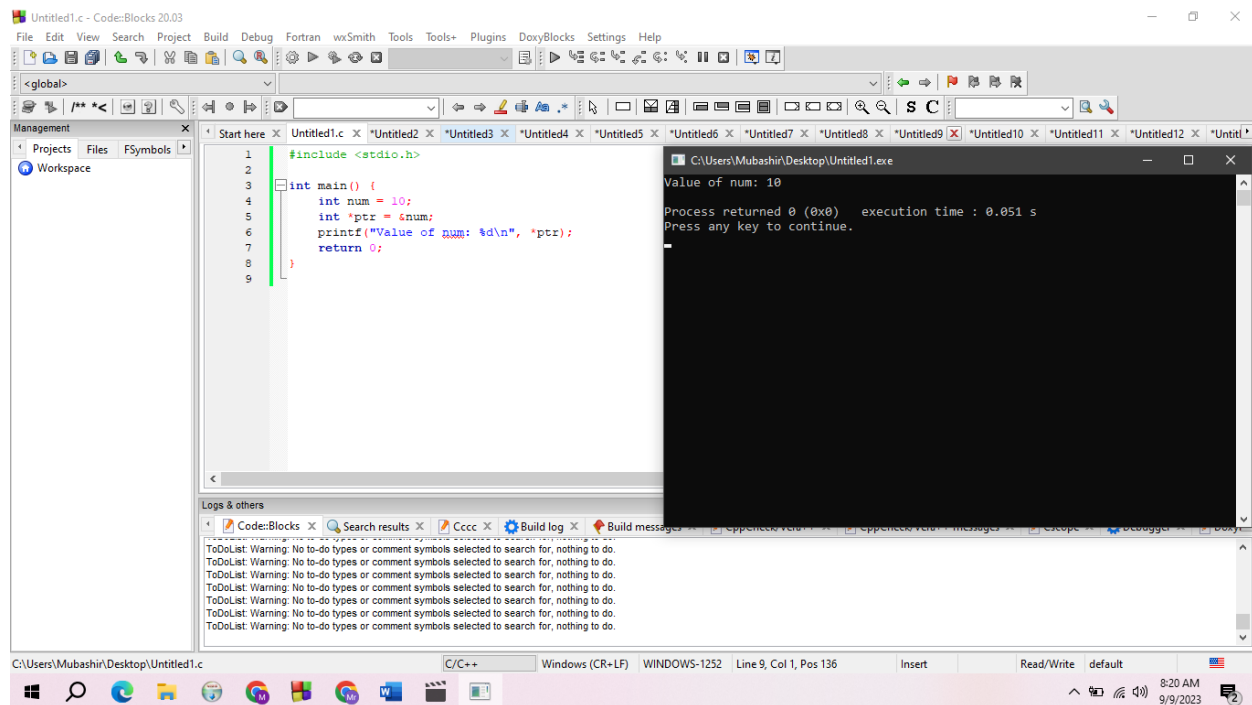
```
    int *ptr = &num;
```

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

```
    return 0;
```

```
}
```

Output:

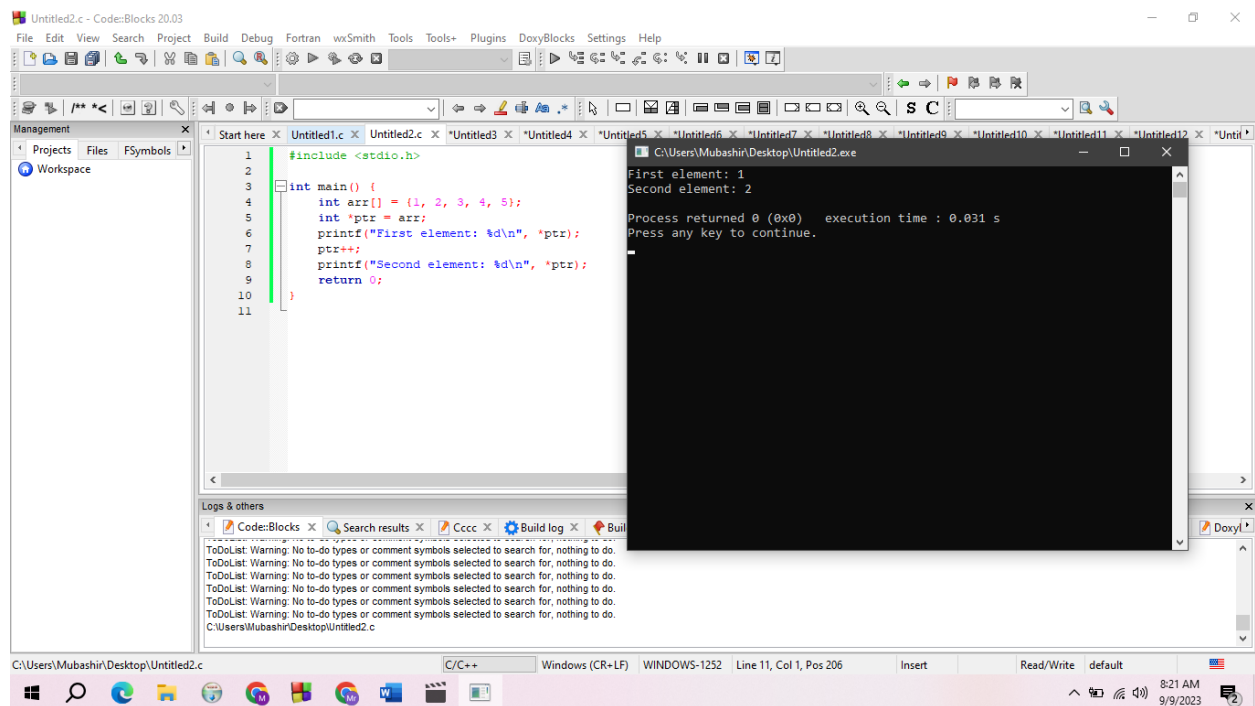


Program # 2

```
#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;
}
```

Output:

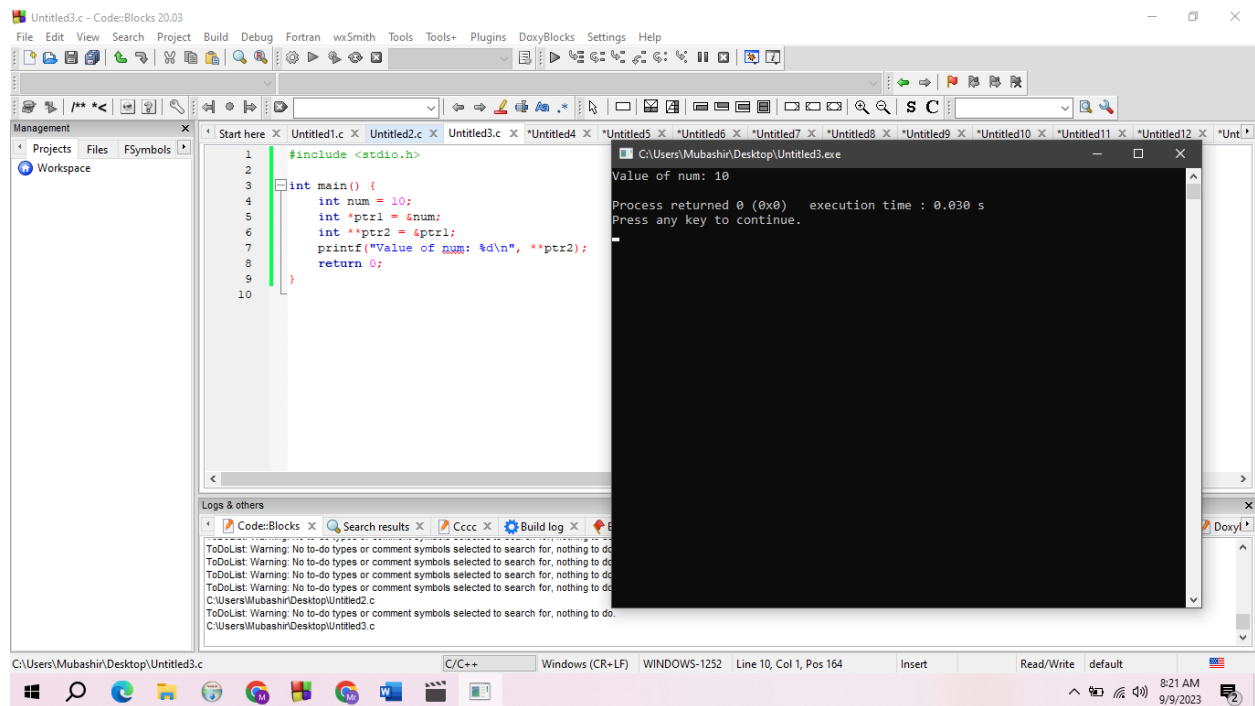


Program # 3

Output:

```
#include <stdio.h>

int main() {
    int num = 10;
    int *ptr1 = &num;
    int **ptr2 = &ptr1;
    printf("Value of num: %d\n", **ptr2);
    return 0;
}
```



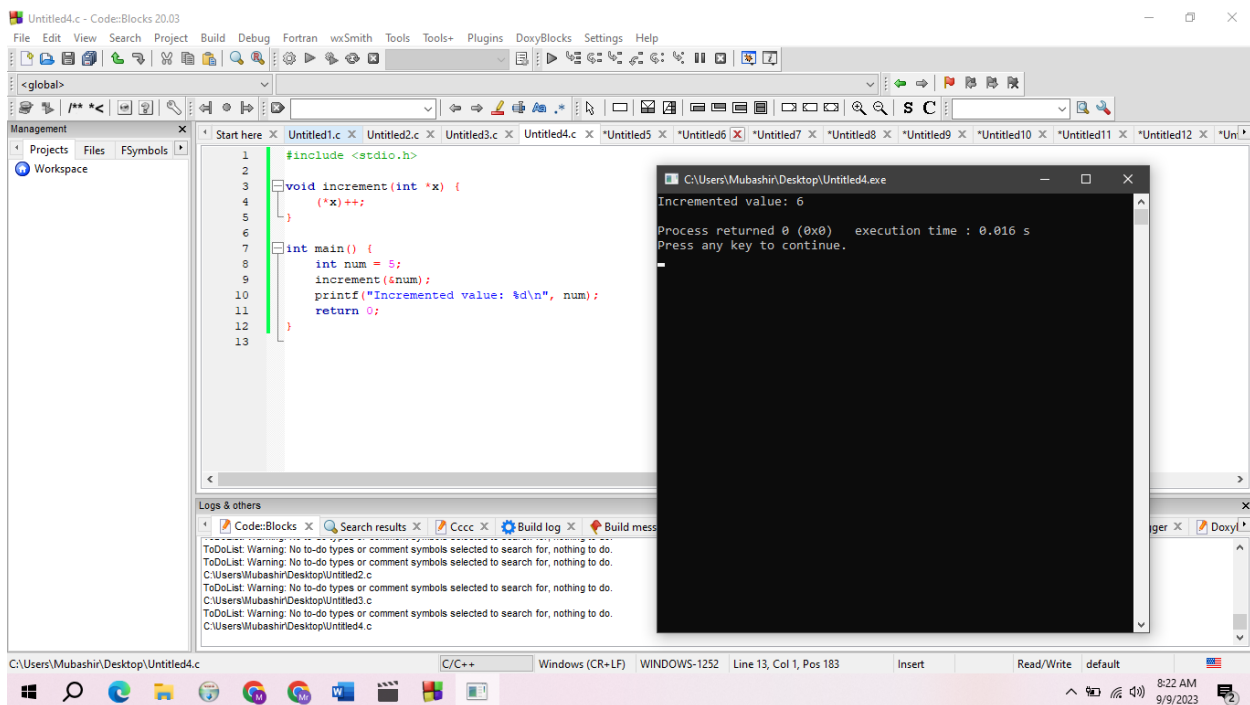
Program # 4

Output:

```
#include <stdio.h>

void increment(int *x) {
    (*x)++;
}

int main() {
    int num = 5;
    increment(&num);
    printf("Incremented value: %d\n", num);
```



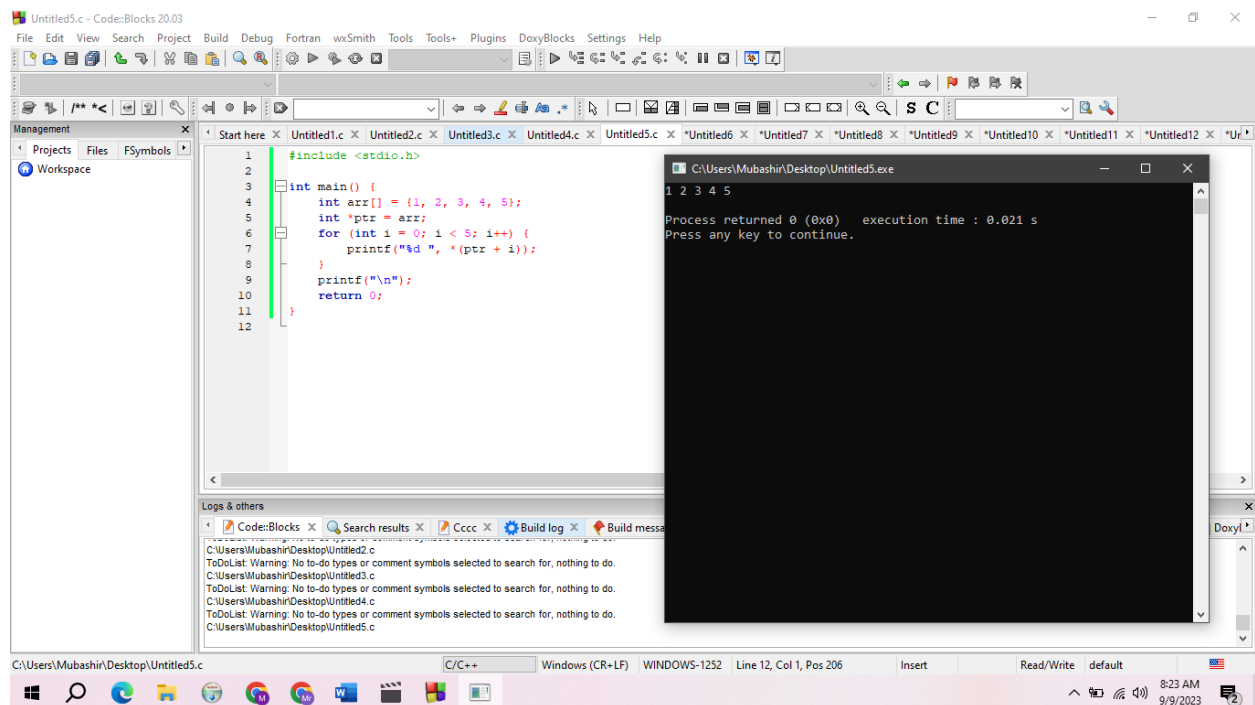
```
    return 0;
}
```

Program # 5

Output:

```
#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;
}
```

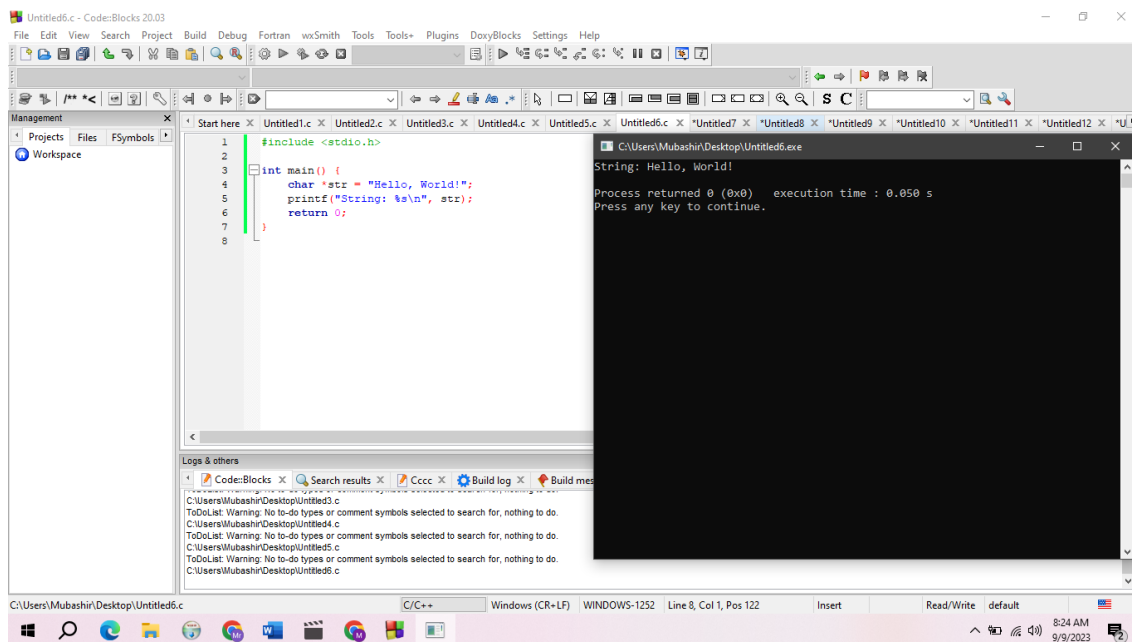


Program # 6

Output:

```
#include <stdio.h>
```

```
int main() {  
    char *str = "Hello, World!";  
    printf("String: %s\n", str);  
    return 0;  
}
```



Program # 7

Output:

```
#include <stdio.h>

#include <stdlib.h>

int main() {

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

    if (ptr != NULL) {

        *ptr = 42;

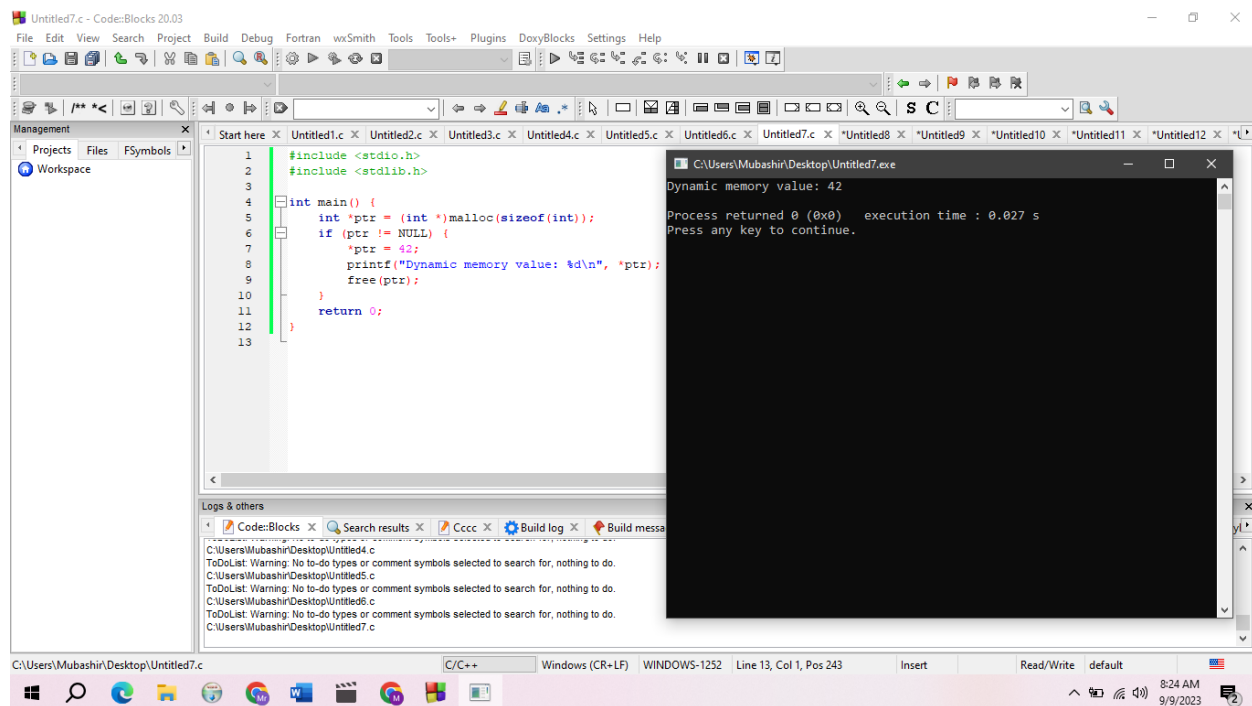
        printf("Dynamic memory value: %d\n", *ptr);

        free(ptr);

    }

    return 0;

}
```



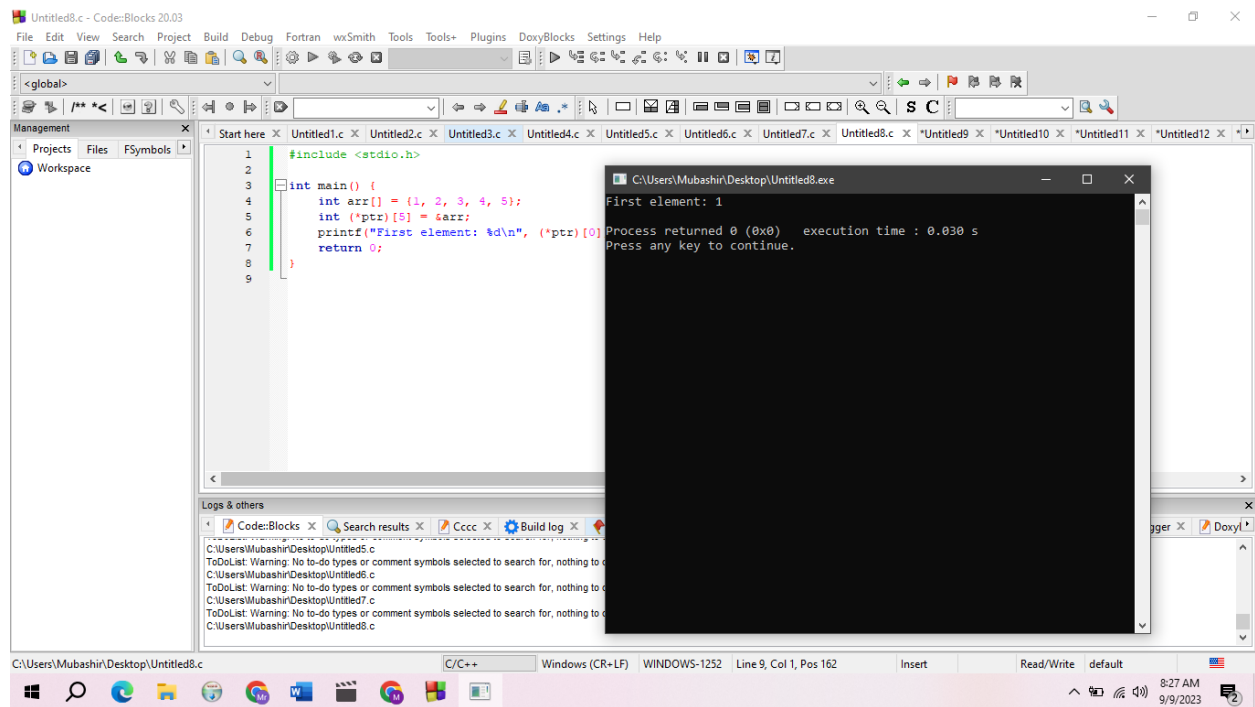
The screenshot displays a code editor window titled 'Untitled7.c - Code::Blocks 20.03'. The editor shows the C program code as defined in the previous block. To the right of the editor, a terminal window titled 'C:\Users\Mubashir\Desktop\Untitled7.exe' shows the program's output: 'Dynamic memory value: 42'. Below the terminal, a 'Logs & others' panel shows build messages, including warnings about no-to-do types or comment symbols selected for search. The Windows taskbar at the bottom indicates the system time is 8:24 AM on 9/9/2023.

Program # 8

Output:

```
#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 # 9

Output:

```
#include <stdio.h>
```

```
int add(int a, int b) {
```

```
    return a + b;
```

```
}
```

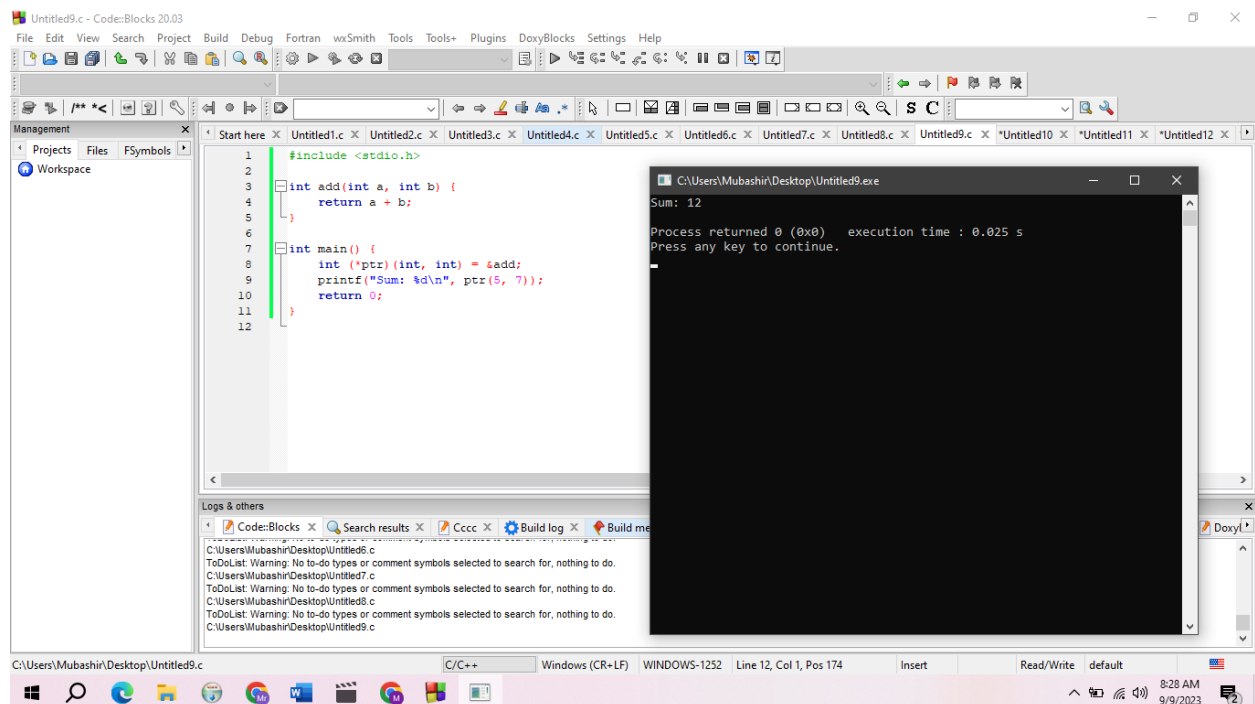
```
int main() {
```

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

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

```
    return 0;
```

```
}
```



Program # 10

Output:

```
#include <stdio.h>

int main() {

    int num1 = 10, num2 = 20;

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

    if (ptr1 == ptr2) {

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

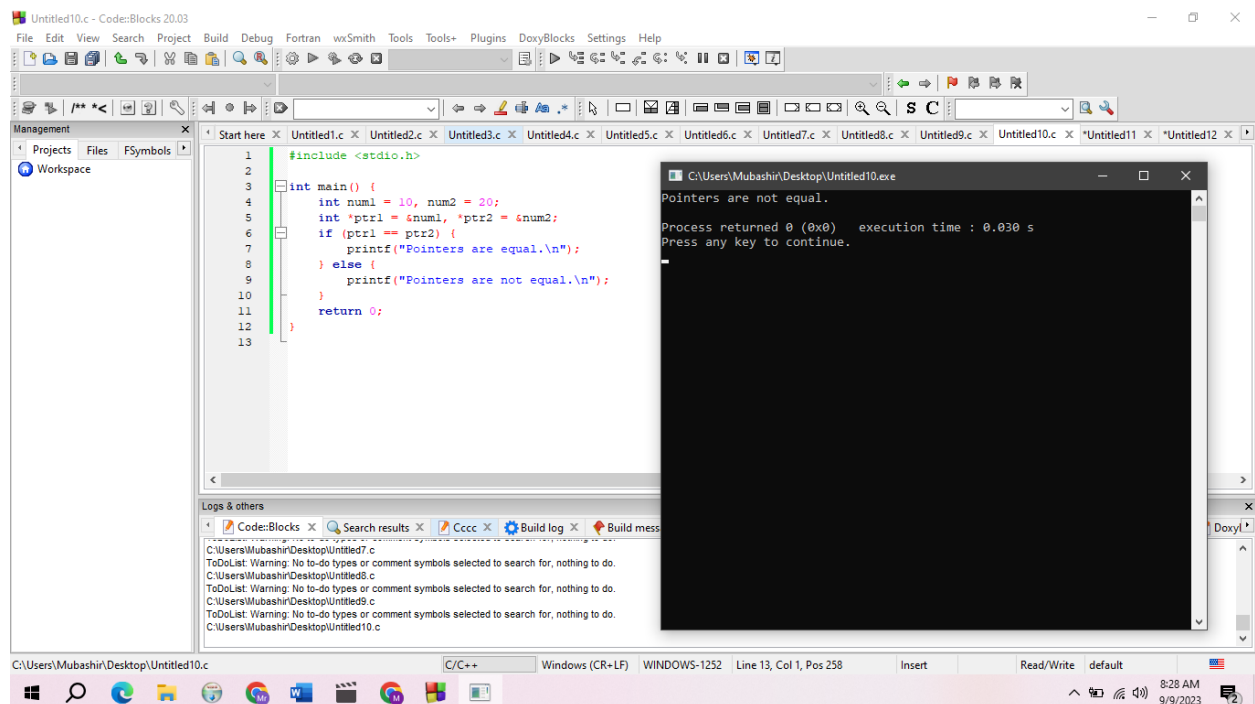
    } else {

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

    }

    return 0;

}
```

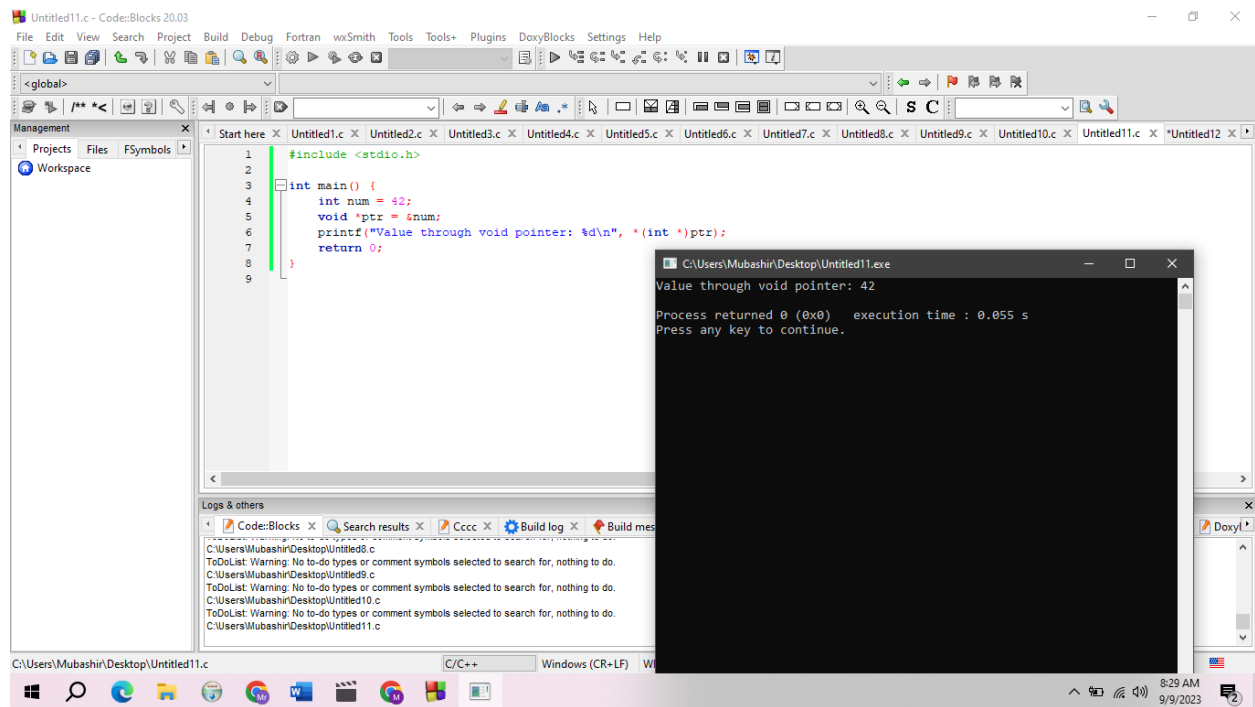


Program # 11

Output:

```
#include <stdio.h>
```

```
int main() {  
    int num = 42;  
    void *ptr = &num;  
    printf("Value through void pointer: %d\n", *(int *)ptr);  
    return 0;  
}
```



Program # 12

Output:

```
#include <stdio.h>
```

```
struct Point {
```

```
    int x, y;
```

```
};
```

```
int main() {
```

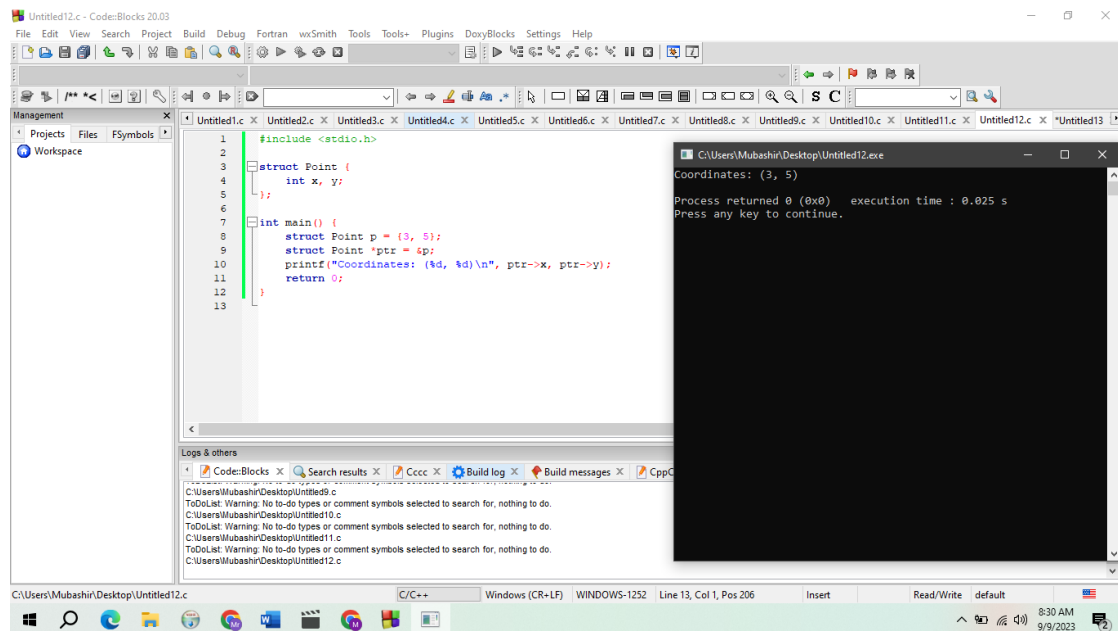
```
    struct Point p = {3, 5};
```

```
    struct Point *ptr = &p;
```

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

```
    return 0;
```

```
}
```



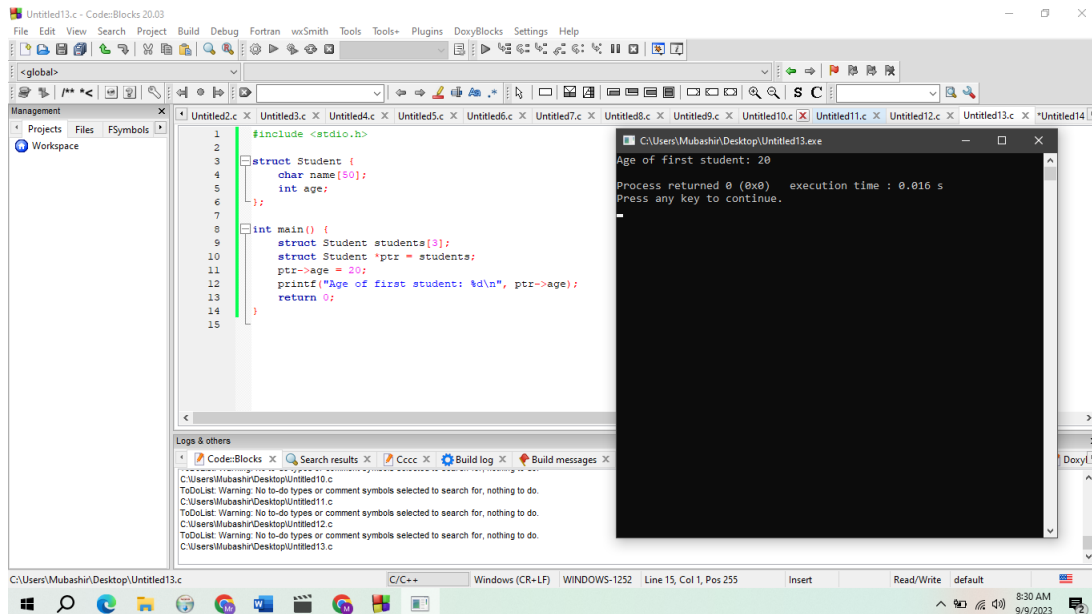
Program # 13

Output:

```
#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

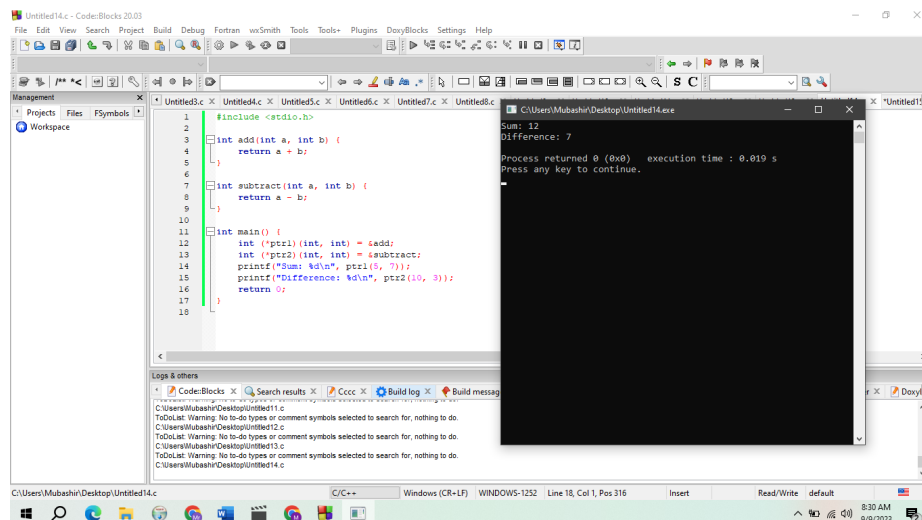
Output:

```
#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

```
#include <stdio.h>

int main() {
    char str[] = "Hello";
    char *ptr = str;
    while (*ptr != '\0') {
        printf("%c ", *ptr);
        ptr++;
    }
    printf("\n");
    return 0;
}
```

Output:

