Day 8 C Program
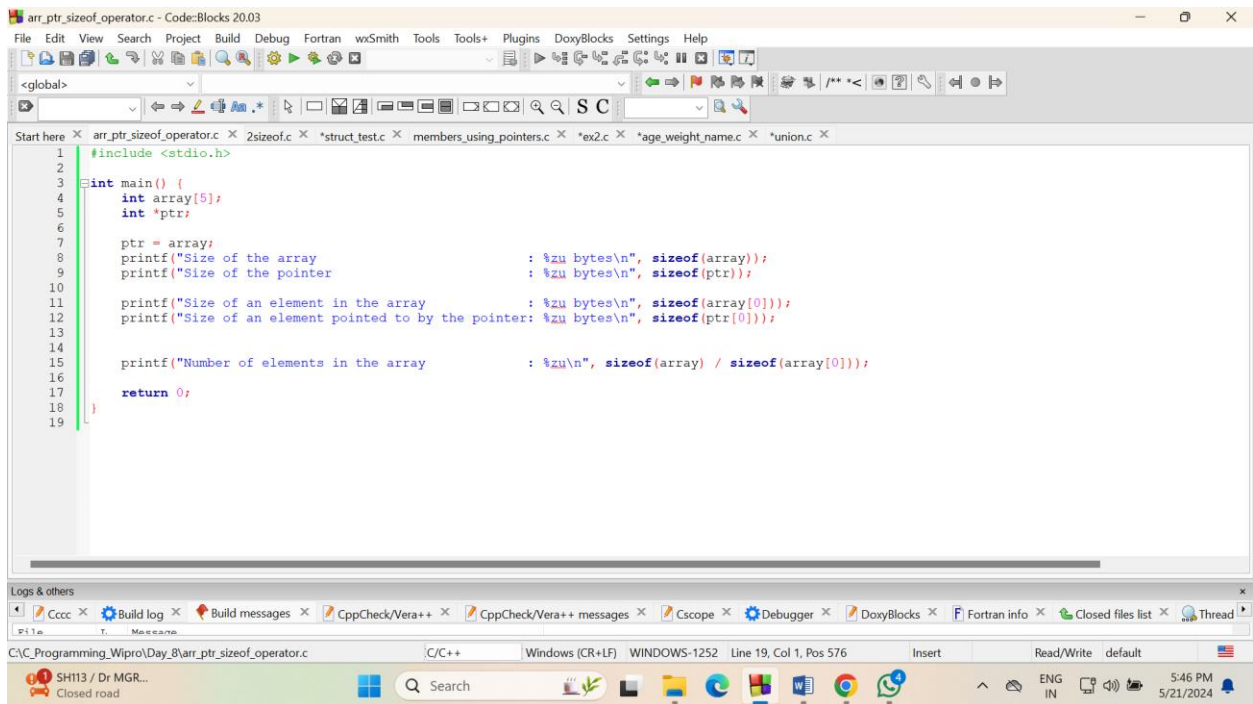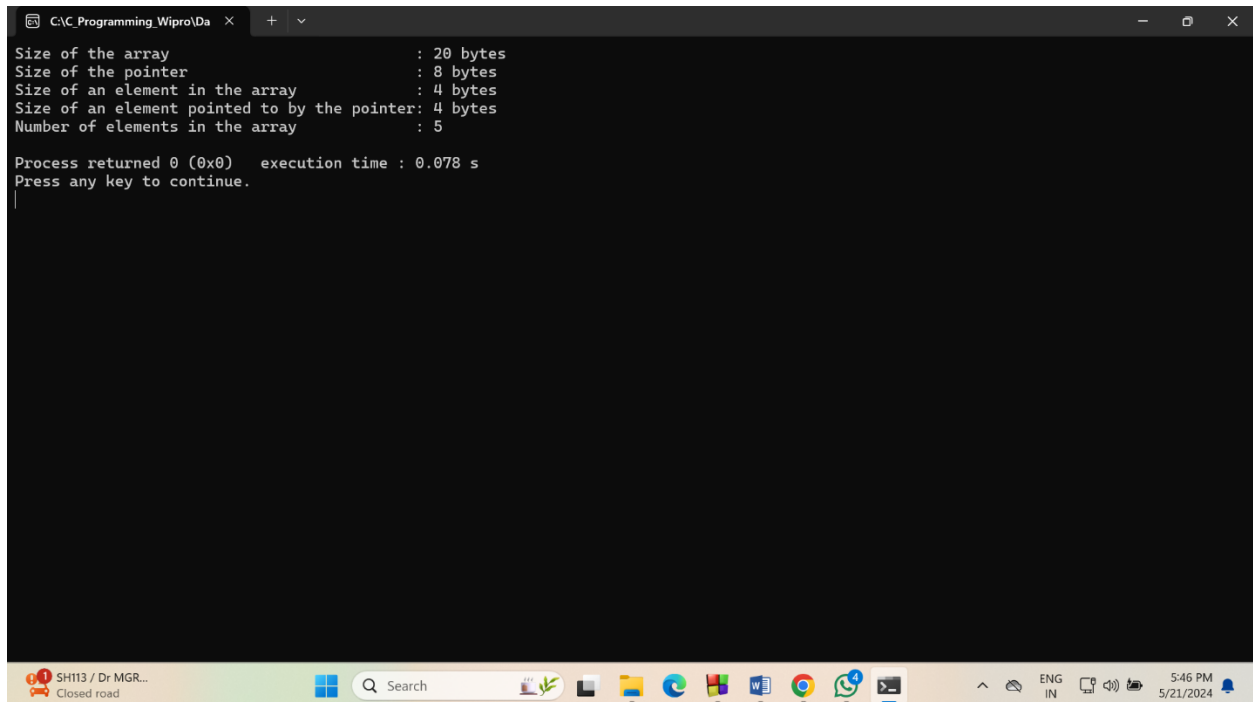
Size of operators:


```c
#include <stdio.h>

int main() {
    int array[5];
    int *ptr;

    ptr = array;
    printf("Size of the array                          : %zu bytes\n", sizeof(array));
    printf("Size of the pointer                        : %zu bytes\n", sizeof(ptr));

    printf("Size of an element in the array            : %zu bytes\n", sizeof(array[0]));
    printf("Size of an element pointed to by the pointer: %zu bytes\n", sizeof(ptr[0]));


    printf("Number of elements in the array            : %zu\n", sizeof(array) / sizeof(array[0]));

    return 0;
}
```
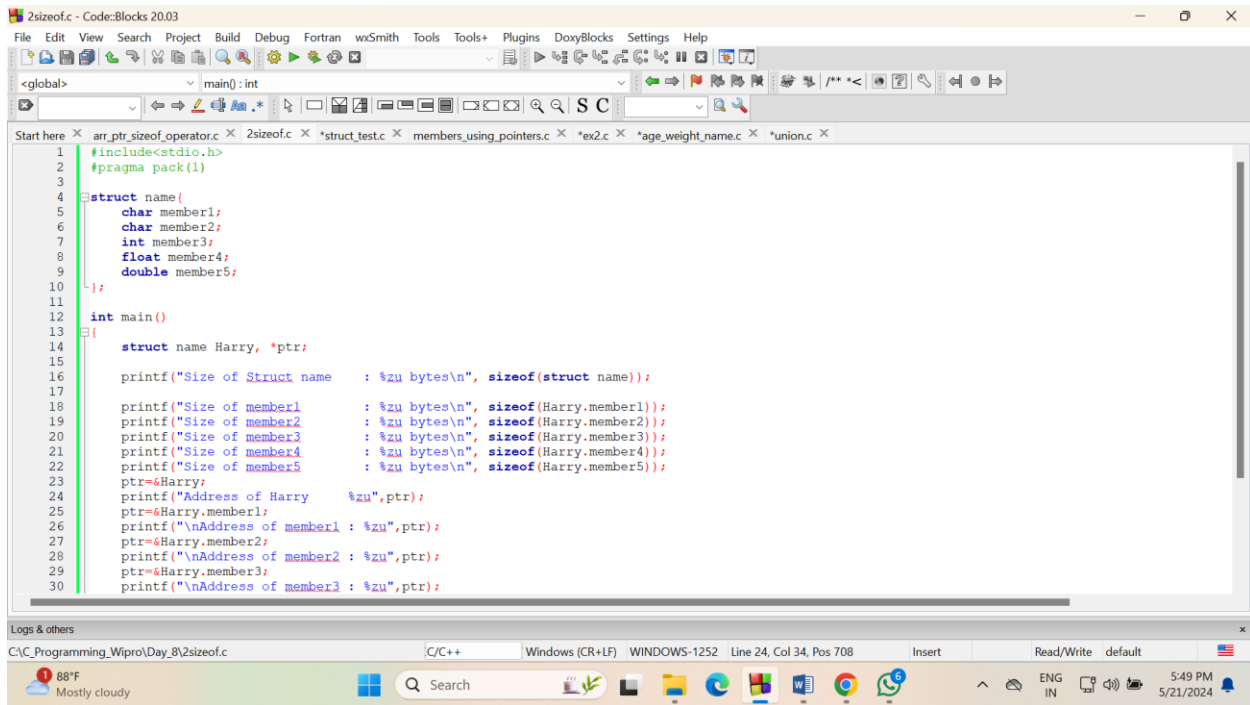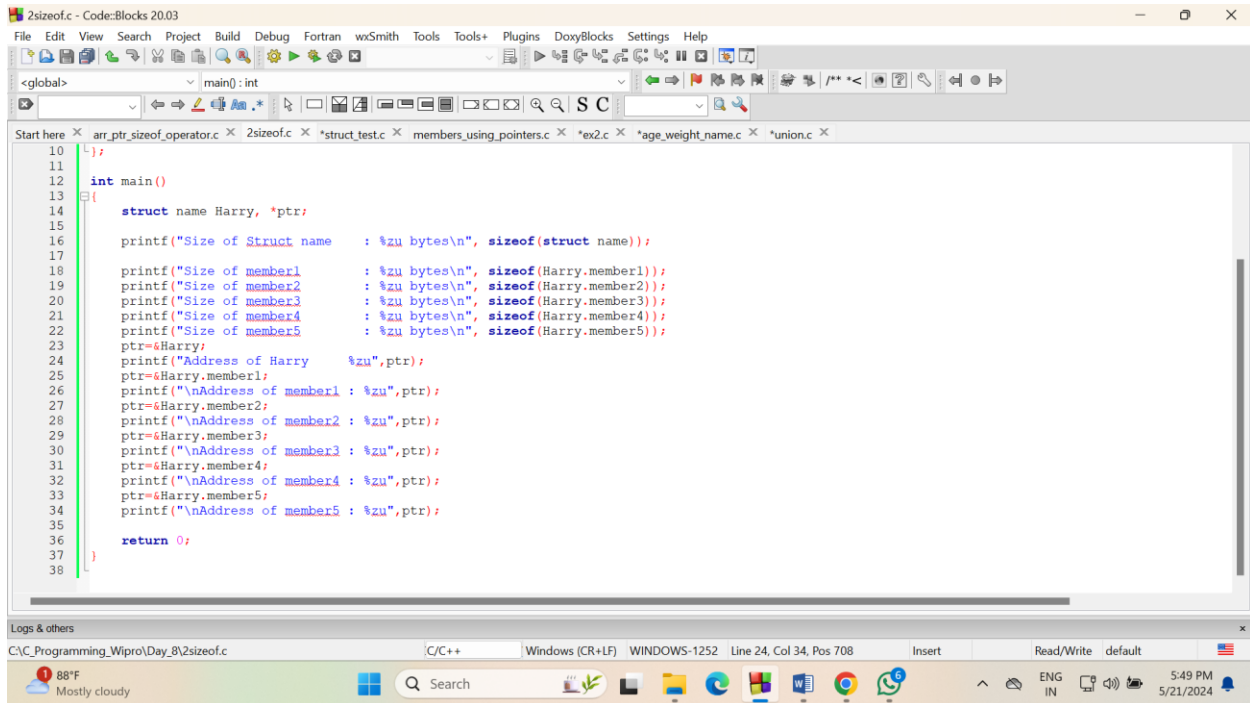
Output:


```
Size of the array                          : 20 bytes
Size of the pointer                        : 8 bytes
Size of an element in the array            : 4 bytes
Size of an element pointed to by the pointer: 4 bytes
Number of elements in the array            : 5

Process returned 0 (0x0)   execution time : 0.078 s
Press any key to continue.
```

Struct padding:


```c
#include<stdio.h>
#pragma pack(1)

struct name{
    char member1;
    char member2;
    int member3;
    float member4;
    double member5;
};

int main()
{
    struct name Harry, *ptr;

    printf("Size of Struct name    : %zu bytes\n", sizeof(struct name));

    printf("Size of member1        : %zu bytes\n", sizeof(Harry.member1));
    printf("Size of member2        : %zu bytes\n", sizeof(Harry.member2));
    printf("Size of member3        : %zu bytes\n", sizeof(Harry.member3));
    printf("Size of member4        : %zu bytes\n", sizeof(Harry.member4));
    printf("Size of member5        : %zu bytes\n", sizeof(Harry.member5));
    ptr=&Harry;
    printf("Address of Harry    %zu",ptr);
    ptr=&Harry.member1;
    printf("\nAddress of member1 : %zu",ptr);
    ptr=&Harry.member2;
    printf("\nAddress of member2 : %zu",ptr);
    ptr=&Harry.member3;
    printf("\nAddress of member3 : %zu",ptr);
```



```c
};

int main()
{
    struct name Harry, *ptr;

    printf("Size of Struct name    : %zu bytes\n", sizeof(struct name));

    printf("Size of member1        : %zu bytes\n", sizeof(Harry.member1));
    printf("Size of member2        : %zu bytes\n", sizeof(Harry.member2));
    printf("Size of member3        : %zu bytes\n", sizeof(Harry.member3));
    printf("Size of member4        : %zu bytes\n", sizeof(Harry.member4));
    printf("Size of member5        : %zu bytes\n", sizeof(Harry.member5));
    ptr=&Harry;
    printf("Address of Harry    %zu",ptr);
    ptr=&Harry.member1;
    printf("\nAddress of member1 : %zu",ptr);
    ptr=&Harry.member2;
    printf("\nAddress of member2 : %zu",ptr);
    ptr=&Harry.member3;
    printf("\nAddress of member3 : %zu",ptr);
    ptr=&Harry.member4;
    printf("\nAddress of member4 : %zu",ptr);
    ptr=&Harry.member5;
    printf("\nAddress of member5 : %zu",ptr);

    return 0;
}
```
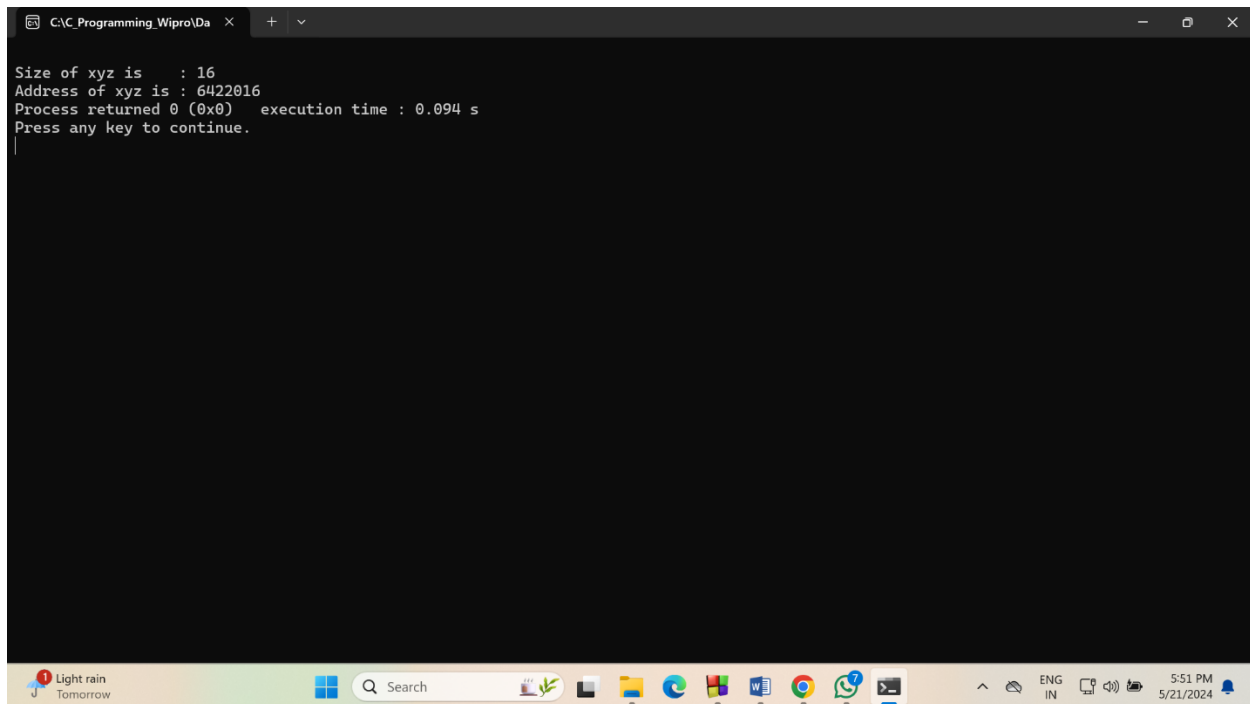
Output:

Size of struct:

Output:

Using StructDisplaying age and weight

Output:

Using pragma print the bytes: