

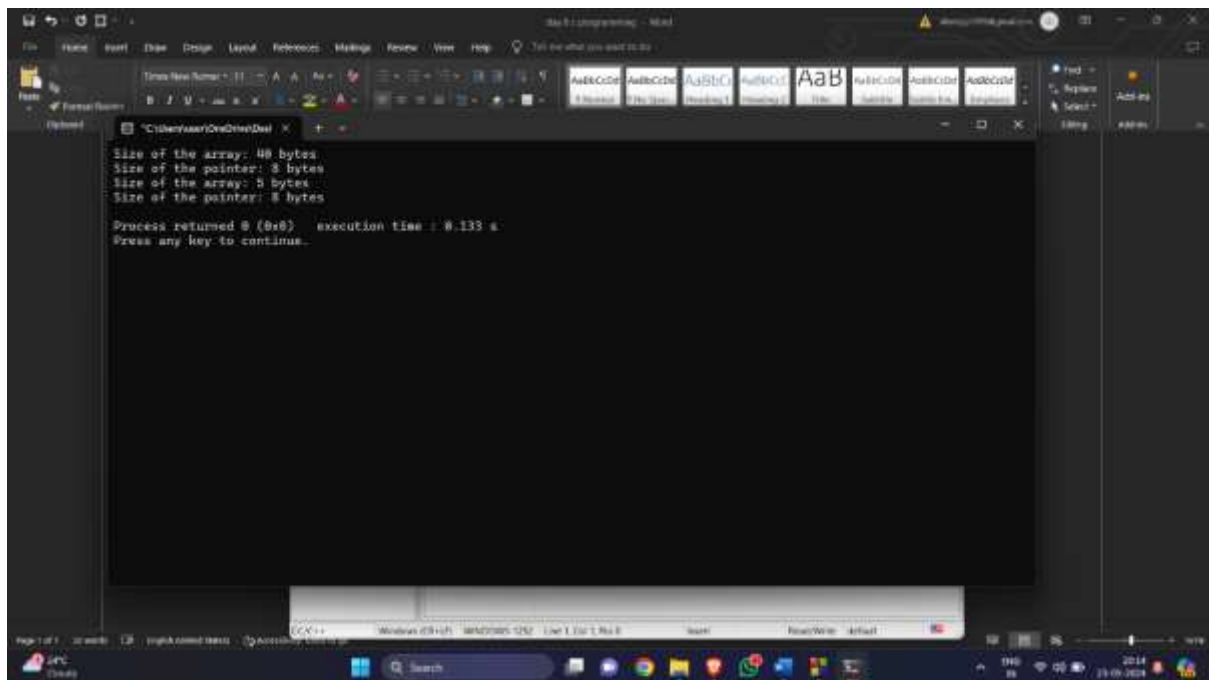
DAY 8

1. Pointer array

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

```
int main() {  
    int array[10];  
    int *ptr;  
    ptr=array;  
    char arr[5];  
    char *pt;  
    pt=arr;  
    printf("Size of the array: %zu bytes\n", sizeof(array));  
    printf("Size of the pointer: %zu bytes\n", sizeof(ptr));  
    printf("Size of the array: %zu bytes\n", sizeof(arr));  
    printf("Size of the pointer: %zu bytes\n", sizeof(pt));  
    return 0;  
}
```

Output:



```
Size of the array: 40 bytes  
Size of the pointer: 8 bytes  
Size of the array: 5 bytes  
Size of the pointer: 8 bytes  
Process returned 0 (0x0)   execution time: 0.133 s  
Press any key to continue...
```

2. Array structure

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

```

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

};

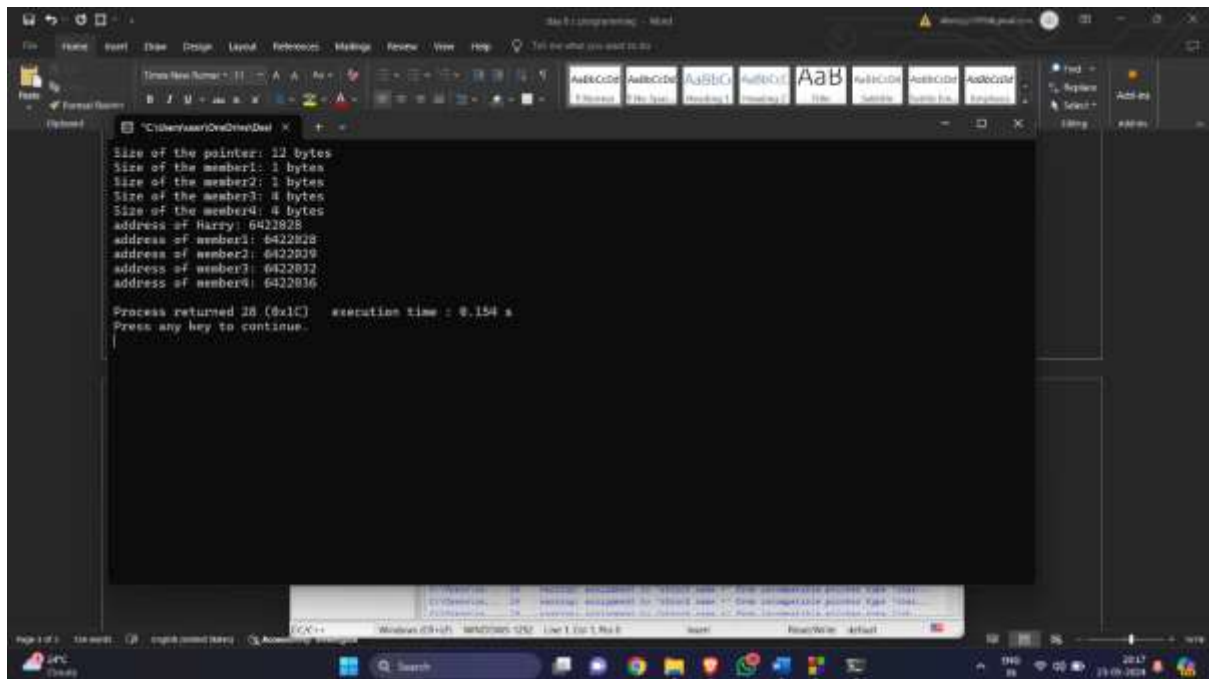
void main()
{
    struct name *ptr, Harry;
    printf("Size of the pointer: %zu bytes\n", sizeof(struct name));
    printf("Size of the member1: %zu bytes\n", sizeof(Harry.member1));
    printf("Size of the member2: %zu bytes\n", sizeof(Harry.member2));
    printf("Size of the member3: %zu bytes\n", sizeof(Harry.member3));
    printf("Size of the member4: %zu bytes\n", sizeof(Harry.member4));
    //printf("Size of the member5: %zu bytes\n", sizeof(Harry.member5));*/

    ptr=&Harry;
    printf("address of Harry: %zu\n",ptr);
    ptr=&Harry.member1;
    printf("address of member1: %zu\n",ptr);
    ptr=&Harry.member2;
    printf("address of member2: %zu\n",ptr);
    ptr=&Harry.member3;
    printf("address of member3: %zu\n",ptr);
    ptr=&Harry.member4;
    printf("address of member4: %zu\n",ptr);
    //ptr=&Harry.member5;
    //printf("address of member5: %zu\n",ptr);*/

```

}

Output:



```
Size of the pointer: 12 bytes
Size of the member1: 1 bytes
Size of the member2: 1 bytes
Size of the member3: 1 bytes
Size of the member4: 1 bytes
address of Harry: 6422828
address of member1: 6422828
address of member2: 6422829
address of member3: 6422832
address of member4: 6422836

Process returned 28 (0x1C)   execution time : 0.154 s
Press any key to continue.
```

3. Function 1

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

```
int addNumbers(int a, int b);    // function prototype
```

```
int main()
```

```
{
```

```
    int n1,n2,sum;
```

```
    printf("Enters two numbers: ");
```

```
    scanf("%d %d",&n1,&n2);
```

```
    sum = addNumbers(n1, n2);    // function call
```

```
    printf("sum = %d",sum);
```

```
    return 0;
```

```
}
```

```
int addNumbers(int a, int b)    // function definition
```

```
{
```

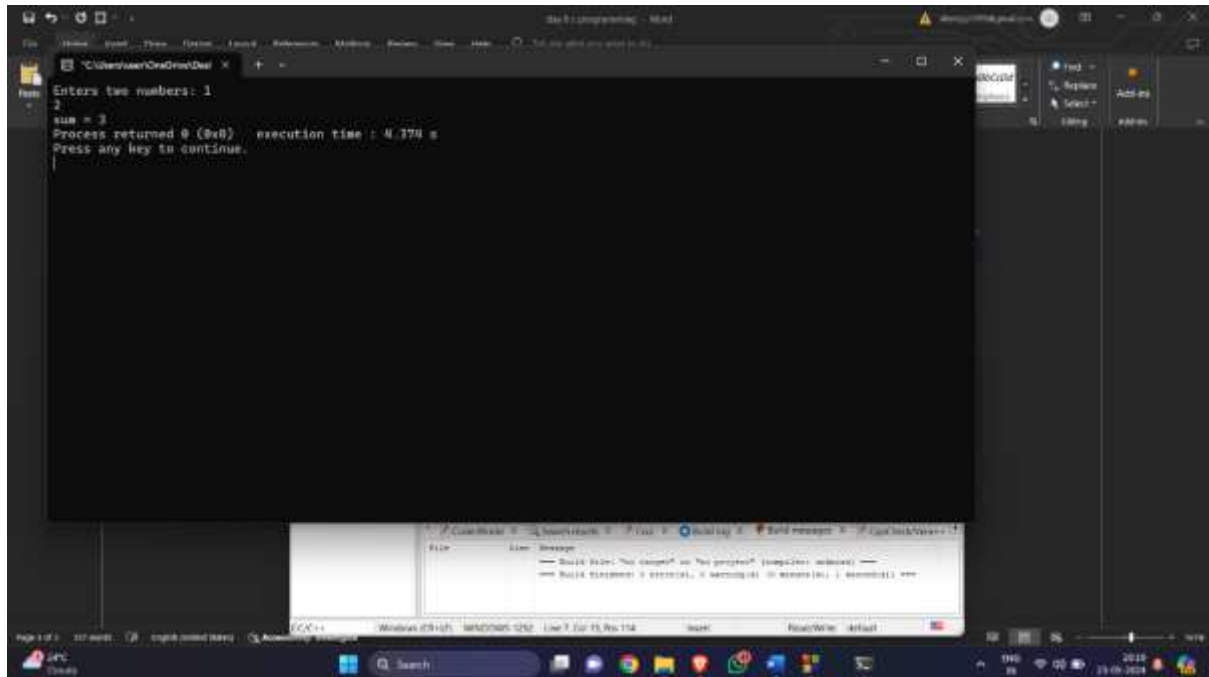
```
    int result;
```

```
    result = a+b;
```

```
    return result;              // return statement
```

}

Output:



4. Function 2

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

```
void checkPrimeNumber();
```

```
int main() {  
    checkPrimeNumber(); // argument is not passed  
    return 0;  
}
```

// return type is void meaning doesn't return any value

```
void checkPrimeNumber() {  
    int n, i, flag = 0;
```

```
    printf("Enter a positive integer: ");  
    scanf("%d",&n);
```

// 0 and 1 are not prime numbers

```
    if (n == 0 || n == 1)  
        flag = 1;
```

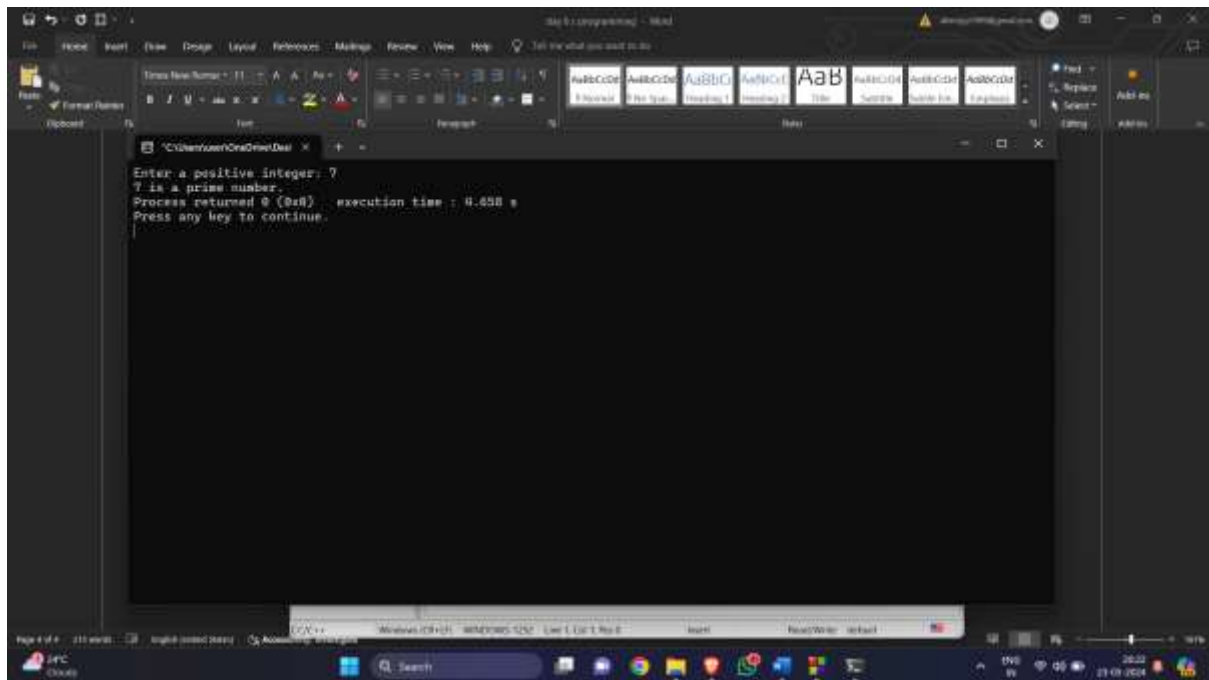
```
    for(i = 2; i <= n/2; ++i) {  
        if(n%i == 0) {  
            flag = 1;  
            break;  
        }  
    }  
}
```

```

if (flag == 1)
    printf("%d is not a prime number.", n);
else
    printf("%d is a prime number.", n);
}

```

Output:



5. Fibonacci

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

```

int fibonacci(int n) {
    if(n == 0)
        return 0;
    else if(n == 1)
        return 1;
    else
        return (fibonacci(n-1) + fibonacci(n-2));
}

```

```

int main() {
    int n;

```

```
printf("Enter the number of terms\n");
```

```
scanf("%d", &n);
```

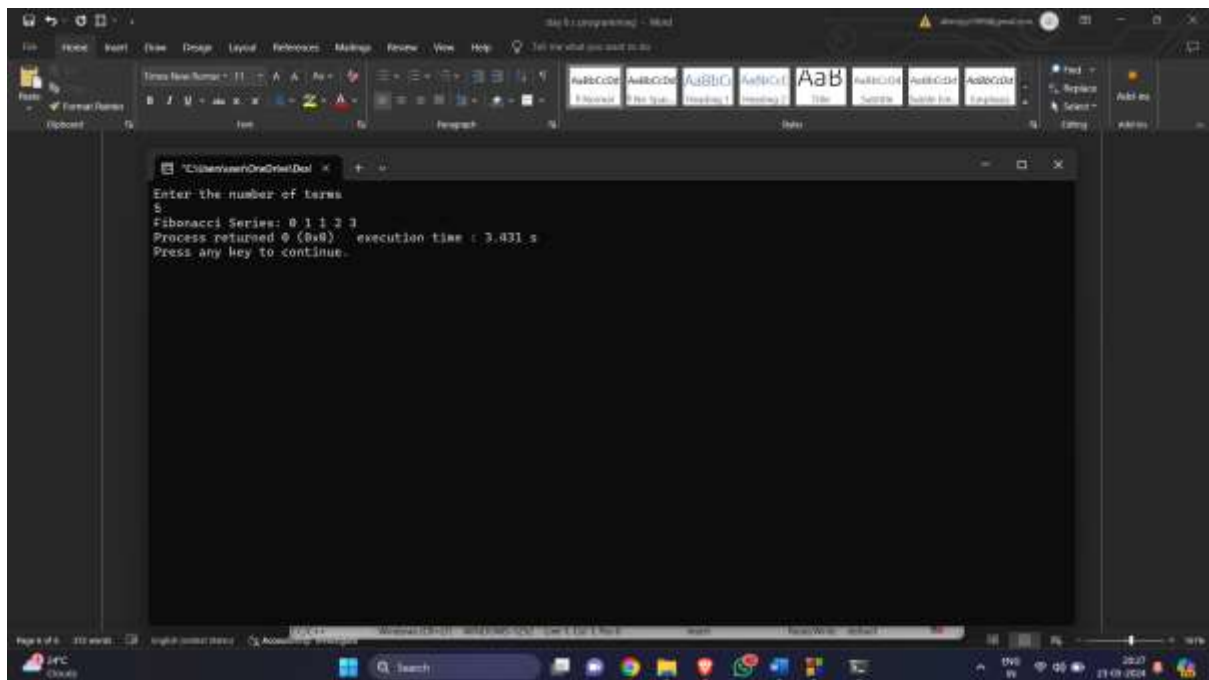
```
printf("Fibonacci Series: ");
```

```
for (int i = 0; i < n; i++) {  
    printf("%d ", fibonacci(i));  
}
```

```
return 0;
```

```
}
```

Output:



The screenshot shows a Windows IDE with a dark theme. The main window displays the output of a C program. The text in the output window is as follows:

```
Enter the number of terms  
5  
Fibonacci Series: 0 1 1 2 3  
Process returned 0 (0x0)   execution time : 3.431 s  
Press any key to continue.
```

The IDE's interface includes a menu bar at the top with options like File, Home, Insert, Table, Design, Layout, References, Markings, Review, View, and Help. Below the menu bar is a ribbon with various toolbars. The bottom of the screen shows the Windows taskbar with the Start button, a search bar, and several pinned application icons. The system tray on the right indicates the date and time as 22/09/2024 and 10:16.