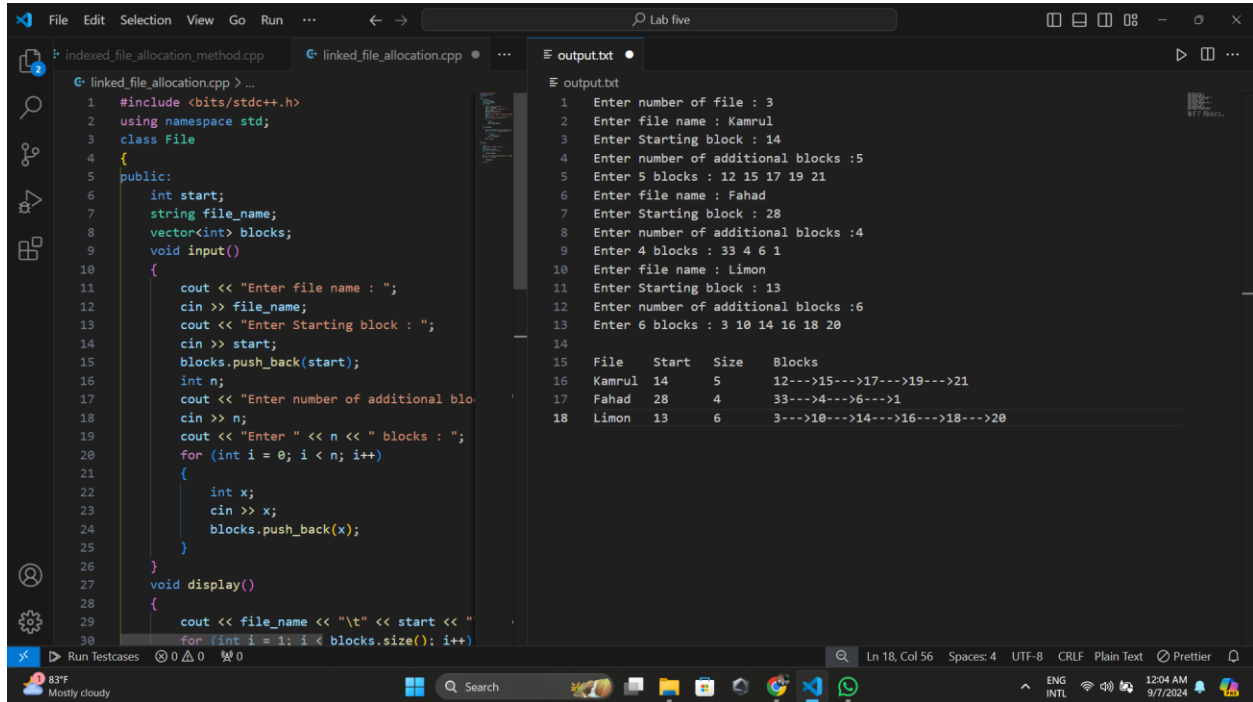


## Output:



The screenshot shows a C++ IDE with two main panels. The left panel displays the source code for `linked_file_allocation.cpp`, and the right panel shows the output in `output.txt`.

**Source Code (`linked_file_allocation.cpp`):**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 class File
4 {
5 public:
6     int start;
7     string file_name;
8     vector<int> blocks;
9     void input()
10    {
11        cout << "Enter file name : ";
12        cin >> file_name;
13        cout << "Enter Starting block : ";
14        cin >> start;
15        blocks.push_back(start);
16        int n;
17        cout << "Enter number of additional blocks : ";
18        cin >> n;
19        cout << "Enter " << n << " blocks : ";
20        for (int i = 0; i < n; i++)
21        {
22            int x;
23            cin >> x;
24            blocks.push_back(x);
25        }
26    }
27    void display()
28    {
29        cout << file_name << "\t" << start << "
30        for (int i = 1; i < blocks.size(); i++)
```

**Output (`output.txt`):**

```
1 Enter number of file : 3
2 Enter file name : Kamrul
3 Enter Starting block : 14
4 Enter number of additional blocks : 5
5 Enter 5 blocks : 12 15 17 19 21
6 Enter file name : Fahad
7 Enter Starting block : 28
8 Enter number of additional blocks : 4
9 Enter 4 blocks : 33 4 6 1
10 Enter file name : Limon
11 Enter Starting block : 13
12 Enter number of additional blocks : 6
13 Enter 6 blocks : 3 10 14 16 18 20
14
15 File    Start    Size    Blocks
16 Kamrul  14      5      12--->15--->17--->19--->21
17 Fahad   28      4      33--->4--->6--->1
18 Limon   13      6      3--->10--->14--->16--->18--->20
```

The output shows the program's execution for three files: Kamrul, Fahad, and Limon. Each file's details (name, starting block, and additional blocks) are entered and then displayed in a structured format.