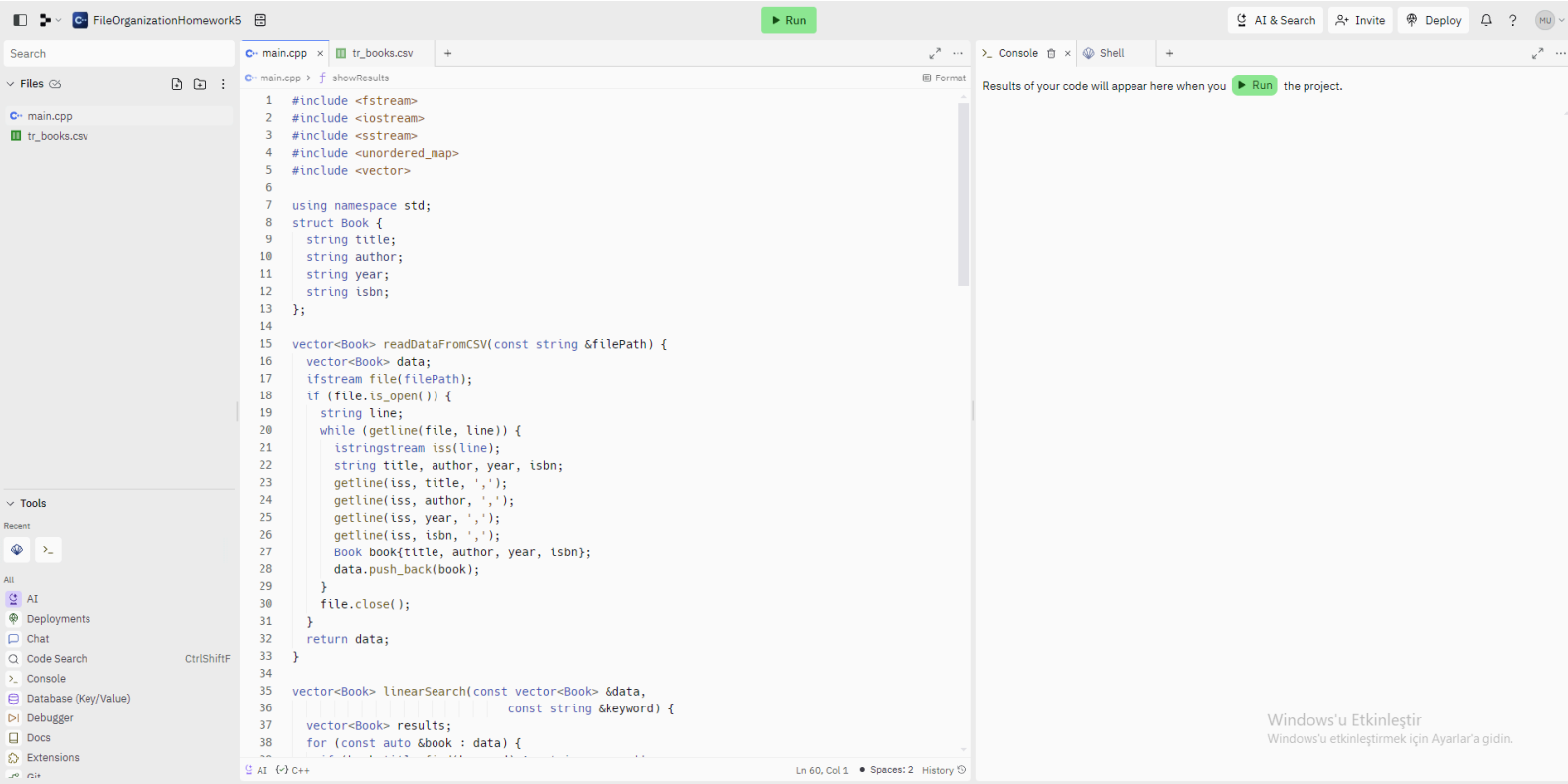


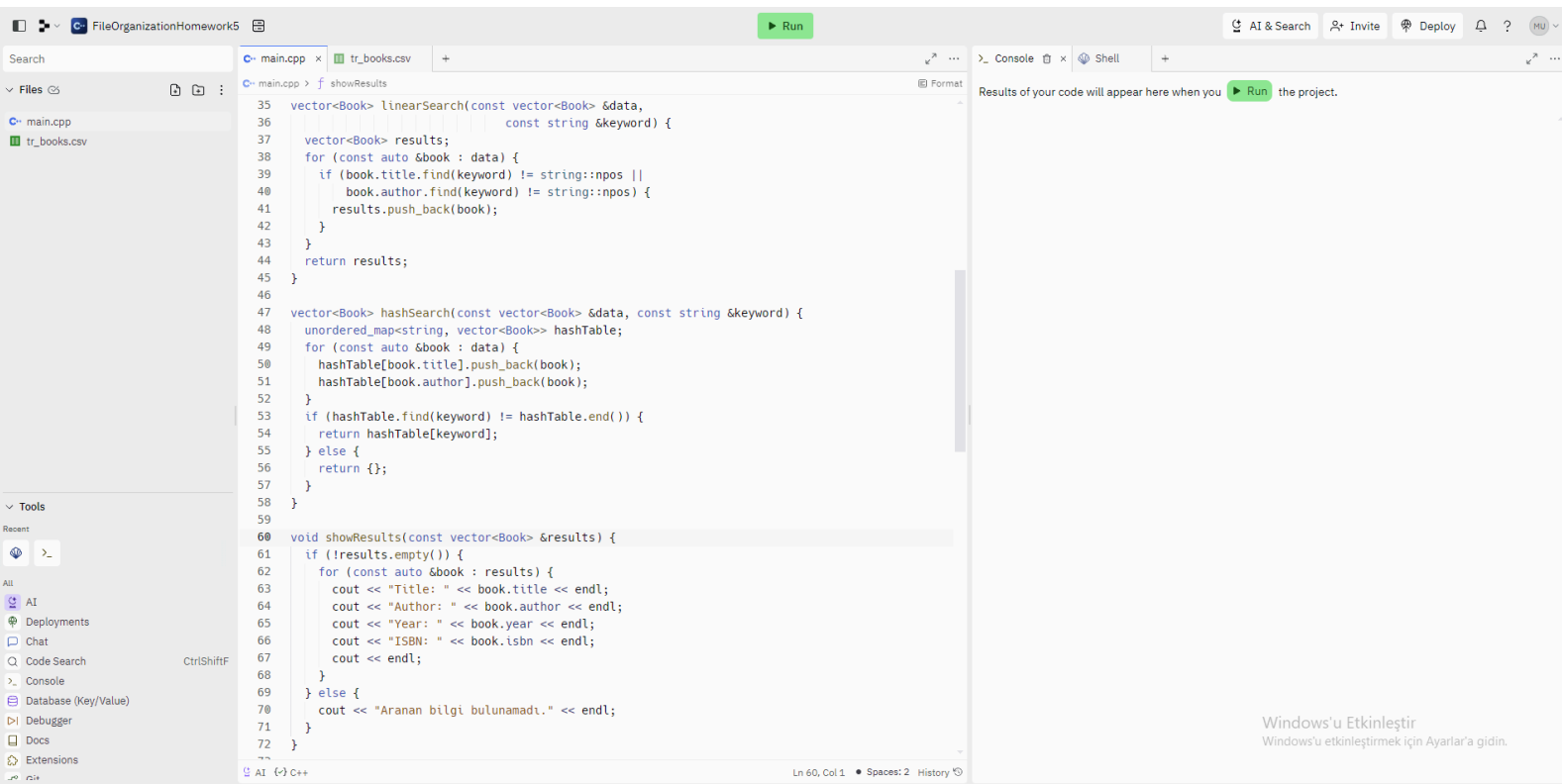
Adı: Mustafa  
Soyadı: Çirci

Numara: 1306210018

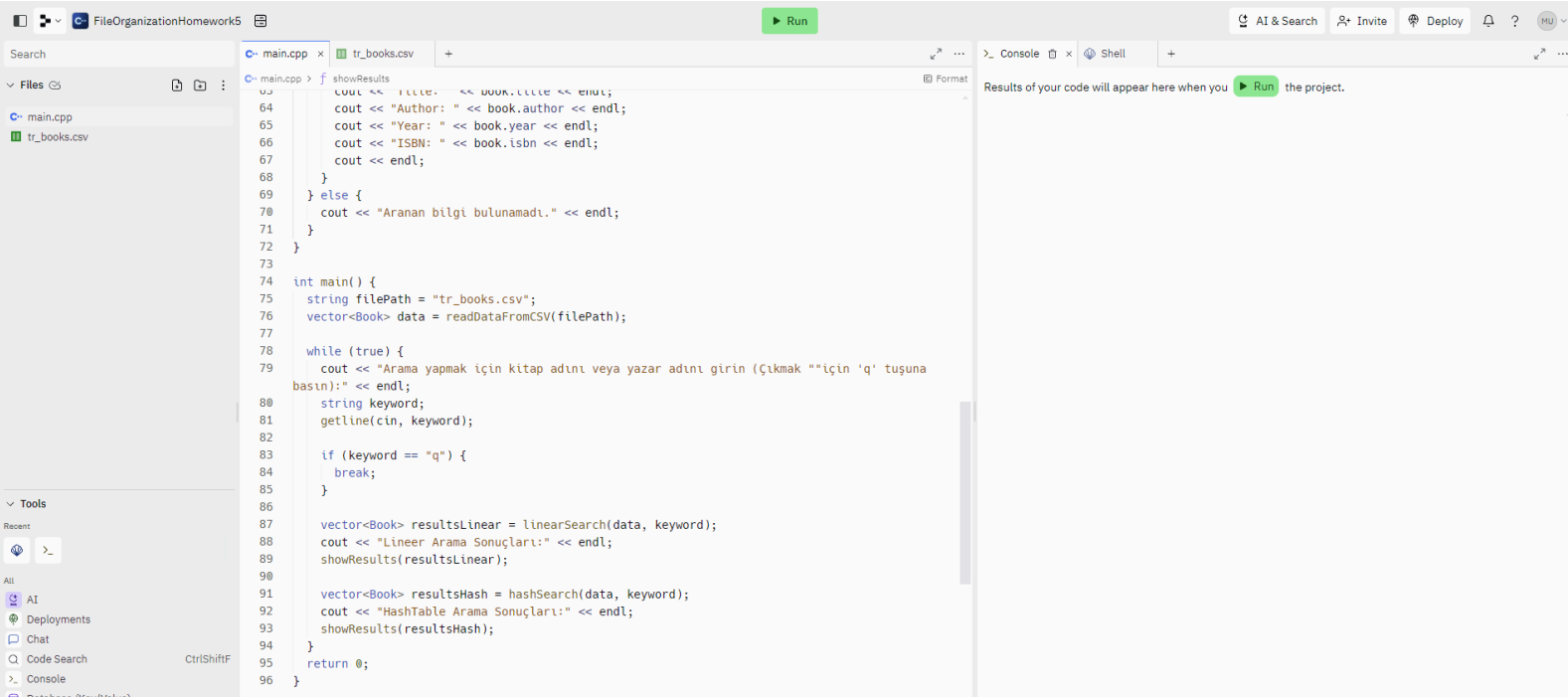
# Lineer Search ve HashTable:



```
1 #include <fstream>
2 #include <iostream>
3 #include <sstream>
4 #include <unordered_map>
5 #include <vector>
6
7 using namespace std;
8 struct Book {
9     string title;
10    string author;
11    string year;
12    string isbn;
13 };
14
15 vector<Book> readDataFromCSV(const string &filePath) {
16     vector<Book> data;
17     ifstream file(filePath);
18     if (file.is_open()) {
19         string line;
20         while (getline(file, line)) {
21             istringstream iss(line);
22             string title, author, year, isbn;
23             getline(iss, title, ',');
24             getline(iss, author, ',');
25             getline(iss, year, ',');
26             getline(iss, isbn, ',');
27             Book book{title, author, year, isbn};
28             data.push_back(book);
29         }
30         file.close();
31     }
32     return data;
33 }
34
35 vector<Book> linearSearch(const vector<Book> &data,
36                           const string &keyword) {
37     vector<Book> results;
38     for (const auto &book : data) {
```



```
39     if (book.title.find(keyword) != string::npos ||
40         book.author.find(keyword) != string::npos) {
41         results.push_back(book);
42     }
43 }
44 return results;
45 }
46
47 vector<Book> hashSearch(const vector<Book> &data, const string &keyword) {
48     unordered_map<string, vector<Book>> hashTable;
49     for (const auto &book : data) {
50         hashTable[book.title].push_back(book);
51         hashTable[book.author].push_back(book);
52     }
53     if (hashTable.find(keyword) != hashTable.end()) {
54         return hashTable[keyword];
55     } else {
56         return {};
57     }
58 }
59
60 void showResults(const vector<Book> &results) {
61     if (!results.empty()) {
62         for (const auto &book : results) {
63             cout << "Title: " << book.title << endl;
64             cout << "Author: " << book.author << endl;
65             cout << "Year: " << book.year << endl;
66             cout << "ISBN: " << book.isbn << endl;
67             cout << endl;
68         }
69     } else {
70         cout << "Aranan bilgi bulunamadı." << endl;
71     }
72 }
```



```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <sstream>
#include <map>
#include <unordered_map>

using namespace std;

struct Book {
    string title;
    string author;
    int year;
    string isbn;
};

vector<Book> readDataFromCSV(string filePath) {
    vector<Book> data;
    ifstream file(filePath);
    if (!file.is_open()) {
        cout << "Dosya bulunamadı." << endl;
        return data;
    }
    string line;
    while (getline(file, line)) {
        stringstream ss(line);
        string title, author, isbn;
        int year;
        ss >> title >> author >> year >> isbn;
        Book book;
        book.title = title;
        book.author = author;
        book.year = year;
        book.isbn = isbn;
        data.push_back(book);
    }
    file.close();
    return data;
}

void showResults(const vector<Book> &data) {
    if (data.empty()) {
        cout << "Aranan bilgi bulunamadı." << endl;
    } else {
        cout << "Title: " << data[0].title << endl;
        cout << "Author: " << data[0].author << endl;
        cout << "Year: " << data[0].year << endl;
        cout << "ISBN: " << data[0].isbn << endl;
    }
}

vector<Book> linearSearch(const vector<Book> &data, string keyword) {
    vector<Book> results;
    for (const Book &book : data) {
        if (book.title.find(keyword) != string::npos) {
            results.push_back(book);
        }
    }
    return results;
}

vector<Book> hashSearch(const vector<Book> &data, string keyword) {
    unordered_map<string, Book> hashTable;
    for (const Book &book : data) {
        hashTable[book.title] = book;
    }
    if (hashTable.find(keyword) != hashTable.end()) {
        return vector<Book> {hashTable[keyword]};
    }
    return vector<Book>();
}

int main() {
    string filePath = "tr_books.csv";
    vector<Book> data = readDataFromCSV(filePath);

    while (true) {
        cout << "Arama yapmak için kitap adını veya yazar adını girin (Çıkmak için 'q' tuşuna basın):" << endl;
        string keyword;
        getline(cin, keyword);

        if (keyword == "q") {
            break;
        }

        vector<Book> resultsLinear = linearSearch(data, keyword);
        cout << "Lineer Arama Sonuçları:" << endl;
        showResults(resultsLinear);

        vector<Book> resultsHash = hashSearch(data, keyword);
        cout << "HashTable Arama Sonuçları:" << endl;
        showResults(resultsHash);
    }

    return 0;
}
```

1. Öncelikle her bir kitap bilgisini tutabilmek için “Book” isminde bir struct oluşturdum. Bu struct yapısının içine başlık, yazar ismi, basım yılı ve ISBN numarasını ekledim.
2. Daha sonrasında dataları CSV dosyasından okumak için “readDataFromCSV” fonksiyonunu oluşturdum. Bu fonksiyon içinde verilen dosya uzantısı ile dosya açıldıktan sonra her satır virgüllerden itibaren ayrılıyor ve “Book” bir obje oluşturuluyor. Her “Book” objesi de “data” ismindeki “Book” vektöründe tutuluyor. Daha sonrasında da bütün kitap verilerini içeren “data” vektörü fonksiyonda return ediliyor.
3. “linearSearch” fonksiyonun içerisinde öncelikle “results” isimli “Book” vektörü oluşturulur. Böylece oluşabilecek olası sonuçlar tek listede tutulur. For döngüsü ile vektörün her bir elemanı gezilirken if bloğunun içinde ilgili keyword’e ait sonuç varsa ilgili “Book” nesnesi result vektörüne eklenir. Daha sonrasında döngü bittikten sonra “results” vektörü return edilir.
4. “hashSearch” fonksiyonun içerisinde “hashTable” isminde bir map oluşturdum. Daha sonrasında for döngüsü ile her bir kitap nesnesini dolaşırken ilgili yazar ve kitap ismini key olarak kullanarak “hashTable” ismindeki map e ekler. Daha sonrasında da map’de bulunan find() fonksiyonu ile ilgili keyword ile arama yaparak sonucu bulur.
5. “showResults” fonksiyonu ile bulunan sonucun ekrana yazdırılması sağlanır.
6. Son olarak “main” fonksiyonu içerisinde de kullanıcılara aramak istedikleri kitap için bir arayüz oluşturularak konsola yazdırılır.