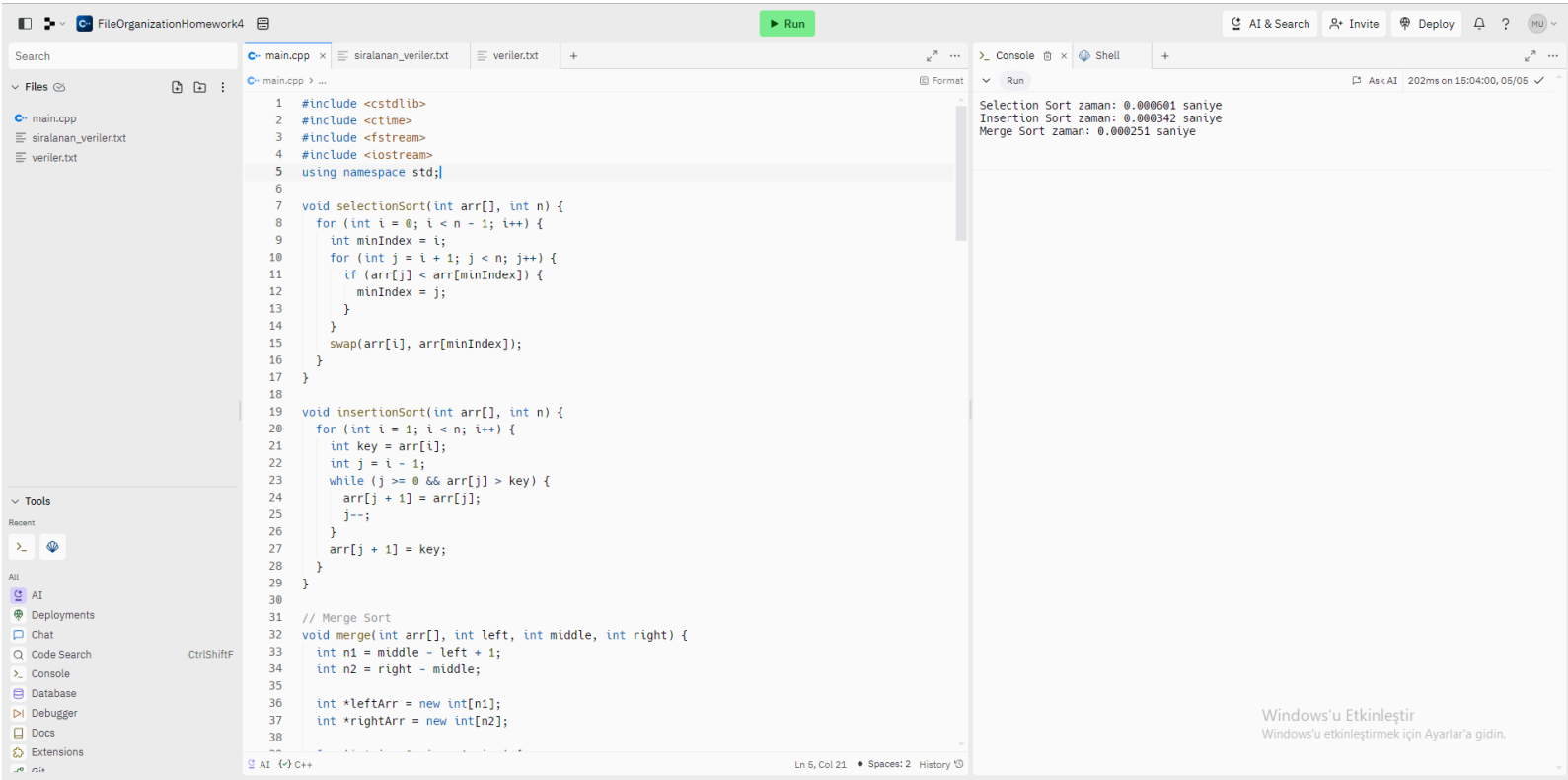


Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018

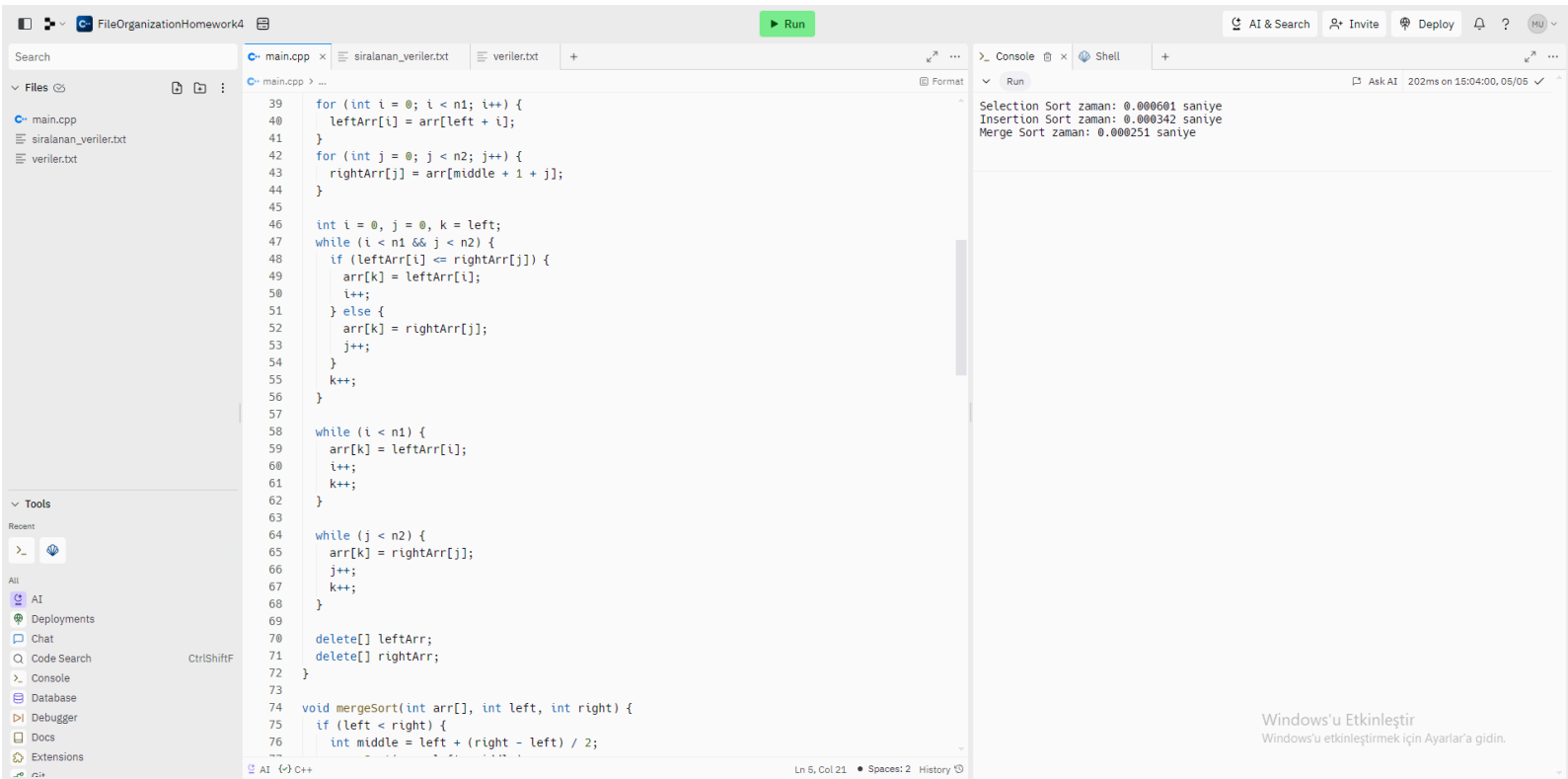
Insertion, Selection ve Merge Sort:



```
1 #include <stdlib>
2 #include <ctime>
3 #include <fstream>
4 #include <iostream>
5 using namespace std;
6
7 void selectionSort(int arr[], int n) {
8     for (int i = 0; i < n - 1; i++) {
9         int minIndex = i;
10        for (int j = i + 1; j < n; j++) {
11            if (arr[j] < arr[minIndex]) {
12                minIndex = j;
13            }
14        }
15        swap(arr[i], arr[minIndex]);
16    }
17 }
18
19 void insertionSort(int arr[], int n) {
20     for (int i = 1; i < n; i++) {
21         int key = arr[i];
22         int j = i - 1;
23         while (j >= 0 && arr[j] > key) {
24             arr[j + 1] = arr[j];
25             j--;
26         }
27         arr[j + 1] = key;
28     }
29 }
30
31 // Merge Sort
32 void merge(int arr[], int left, int middle, int right) {
33     int n1 = middle - left + 1;
34     int n2 = right - middle;
35
36     int *leftArr = new int[n1];
37     int *rightArr = new int[n2];
38
39     for (int i = 0; i < n1; i++) {
40         leftArr[i] = arr[left + i];
41     }
42     for (int j = 0; j < n2; j++) {
43         rightArr[j] = arr[middle + 1 + j];
44     }
45
46     int i = 0, j = 0, k = left;
47     while (i < n1 && j < n2) {
48         if (leftArr[i] <= rightArr[j]) {
49             arr[k] = leftArr[i];
50             i++;
51         } else {
52             arr[k] = rightArr[j];
53             j++;
54         }
55         k++;
56     }
57
58     while (i < n1) {
59         arr[k] = leftArr[i];
60         i++;
61         k++;
62     }
63
64     while (j < n2) {
65         arr[k] = rightArr[j];
66         j++;
67         k++;
68     }
69
70     delete[] leftArr;
71     delete[] rightArr;
72 }
73
74 void mergeSort(int arr[], int left, int right) {
75     if (left < right) {
76         int middle = left + (right - left) / 2;
77         mergeSort(arr, left, middle);
78         mergeSort(arr, middle + 1, right);
79         merge(arr, left, middle, right);
80     }
81 }
```

Selection Sort zaman: 0.000601 saniye
Insertion Sort zaman: 0.000342 saniye
Merge Sort zaman: 0.000251 saniye

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.



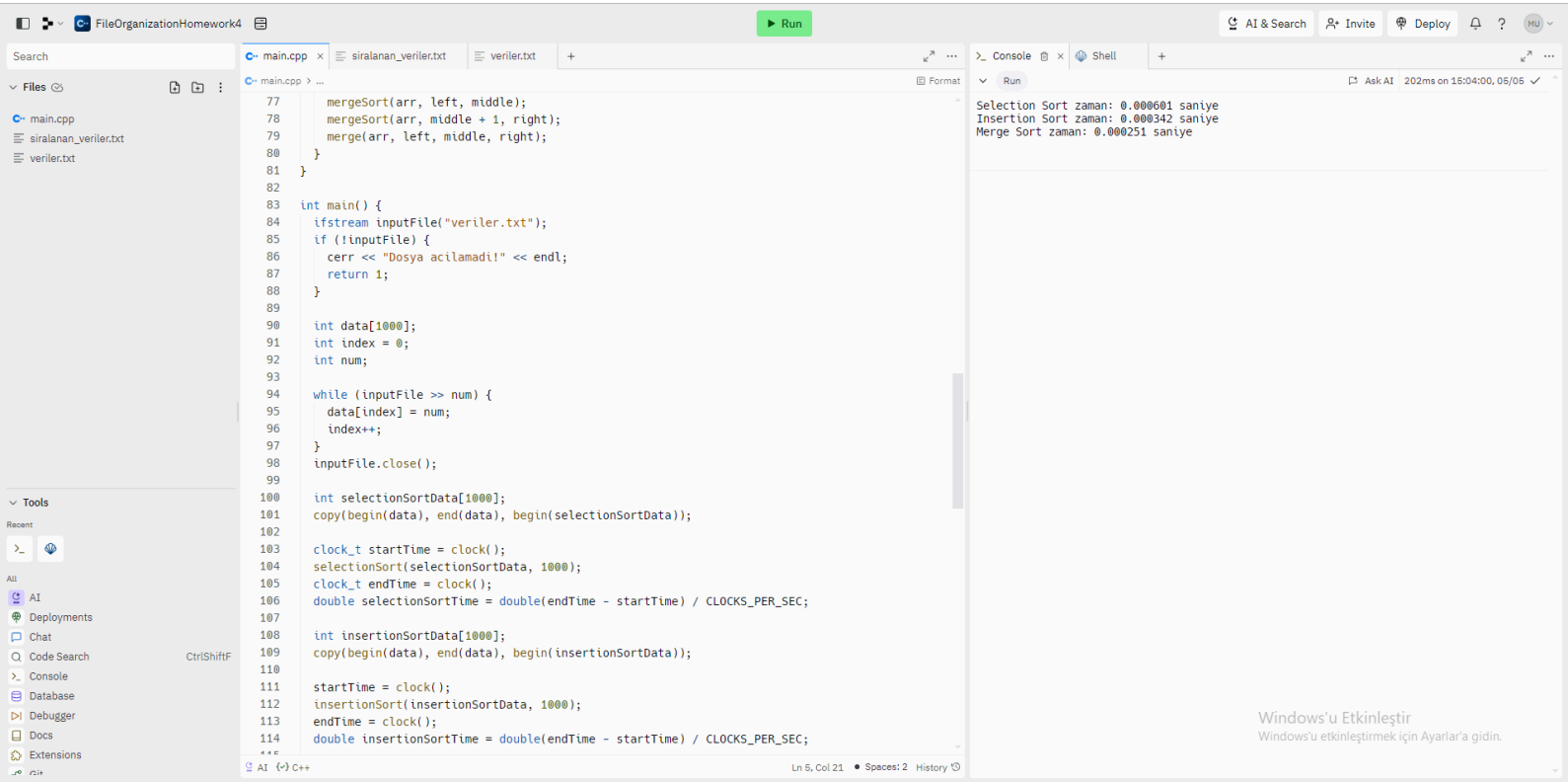
```
39     for (int i = 0; i < n1; i++) {
40         leftArr[i] = arr[left + i];
41     }
42     for (int j = 0; j < n2; j++) {
43         rightArr[j] = arr[middle + 1 + j];
44     }
45
46     int i = 0, j = 0, k = left;
47     while (i < n1 && j < n2) {
48         if (leftArr[i] <= rightArr[j]) {
49             arr[k] = leftArr[i];
50             i++;
51         } else {
52             arr[k] = rightArr[j];
53             j++;
54         }
55         k++;
56     }
57
58     while (i < n1) {
59         arr[k] = leftArr[i];
60         i++;
61         k++;
62     }
63
64     while (j < n2) {
65         arr[k] = rightArr[j];
66         j++;
67         k++;
68     }
69
70     delete[] leftArr;
71     delete[] rightArr;
72 }
73
74 void mergeSort(int arr[], int left, int right) {
75     if (left < right) {
76         int middle = left + (right - left) / 2;
77         mergeSort(arr, left, middle);
78         mergeSort(arr, middle + 1, right);
79         merge(arr, left, middle, right);
80     }
81 }
```

Selection Sort zaman: 0.000601 saniye
Insertion Sort zaman: 0.000342 saniye
Merge Sort zaman: 0.000251 saniye

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

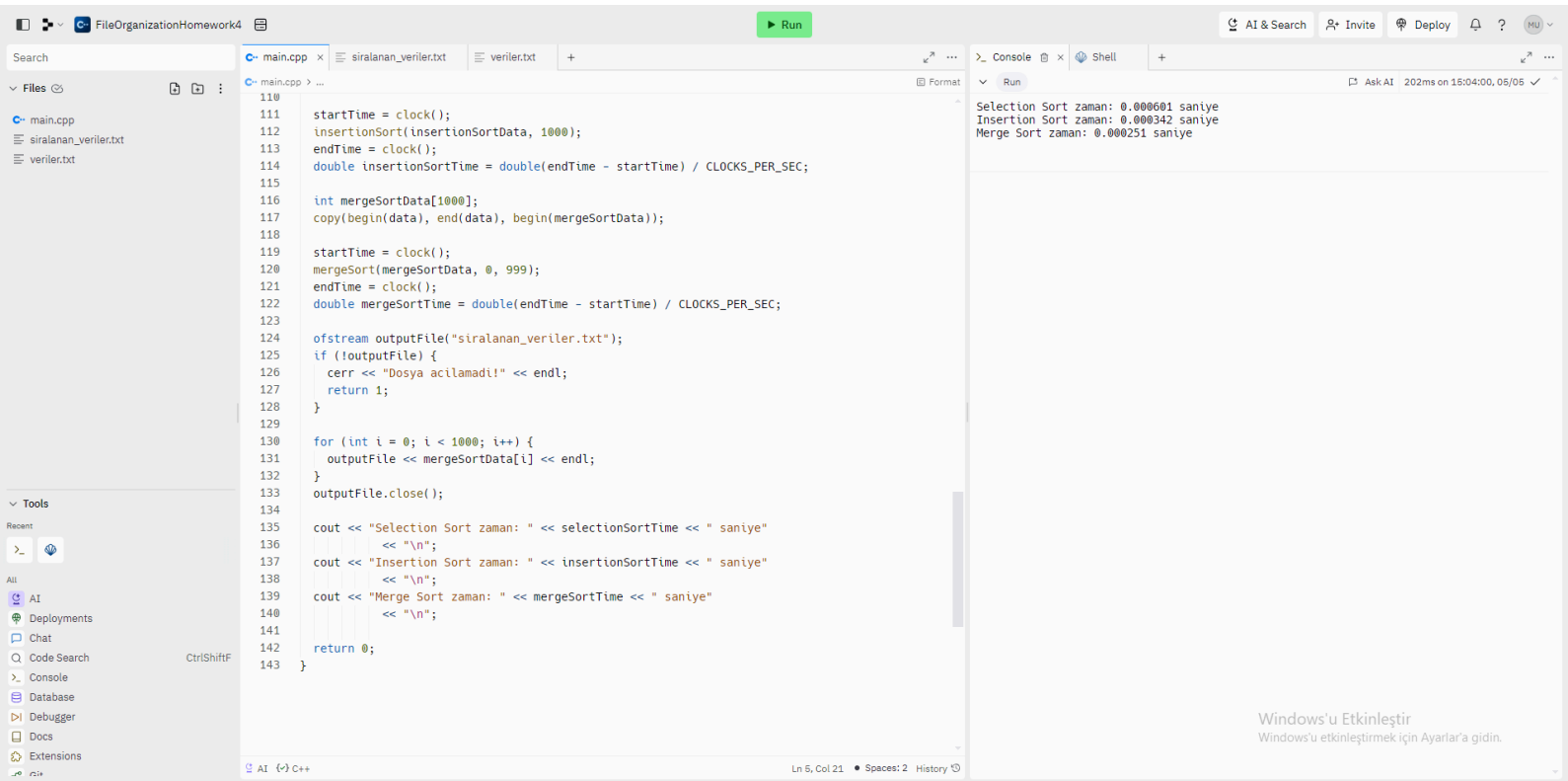
Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018



```
77     mergeSort(arr, left, middle);
78     mergeSort(arr, middle + 1, right);
79     merge(arr, left, middle, right);
80 }
81 }
82
83 int main() {
84     ifstream inputFile("veriler.txt");
85     if (!inputFile) {
86         cerr << "Dosya acilamadi!" << endl;
87         return 1;
88     }
89
90     int data[1000];
91     int index = 0;
92     int num;
93
94     while (inputFile >> num) {
95         data[index] = num;
96         index++;
97     }
98     inputFile.close();
99
100    int selectionSortData[1000];
101    copy(begin(data), end(data), begin(selectionSortData));
102
103    clock_t startTime = clock();
104    selectionSort(selectionSortData, 1000);
105    clock_t endTime = clock();
106    double selectionSortTime = double(endTime - startTime) / CLOCKS_PER_SEC;
107
108    int insertionSortData[1000];
109    copy(begin(data), end(data), begin(insertionSortData));
110
111    startTime = clock();
112    insertionSort(insertionSortData, 1000);
113    endTime = clock();
114    double insertionSortTime = double(endTime - startTime) / CLOCKS_PER_SEC;
```

Selection Sort zaman: 0.000601 saniye
Insertion Sort zaman: 0.000342 saniye
Merge Sort zaman: 0.000251 saniye



```
110
111    startTime = clock();
112    insertionSort(insertionSortData, 1000);
113    endTime = clock();
114    double insertionSortTime = double(endTime - startTime) / CLOCKS_PER_SEC;
115
116    int mergeSortData[1000];
117    copy(begin(data), end(data), begin(mergeSortData));
118
119    startTime = clock();
120    mergeSort(mergeSortData, 0, 999);
121    endTime = clock();
122    double mergeSortTime = double(endTime - startTime) / CLOCKS_PER_SEC;
123
124    ofstream outputFile("siralanan_veriler.txt");
125    if (!outputFile) {
126        cerr << "Dosya acilamadi!" << endl;
127        return 1;
128    }
129
130    for (int i = 0; i < 1000; i++) {
131        outputFile << mergeSortData[i] << endl;
132    }
133    outputFile.close();
134
135    cout << "Selection Sort zaman: " << selectionSortTime << " saniye"
136         << "\n";
137    cout << "Insertion Sort zaman: " << insertionSortTime << " saniye"
138         << "\n";
139    cout << "Merge Sort zaman: " << mergeSortTime << " saniye"
140         << "\n";
141
142    return 0;
143 }
```

Selection Sort zaman: 0.000601 saniye
Insertion Sort zaman: 0.000342 saniye
Merge Sort zaman: 0.000251 saniye

Adı: Mustafa
Soyadı: Çirci

Numara: 1306210018

1. Öncelikle Selection sort, Insertion sort ve Merge sort için gerekli kodları yazdım.
2. Daha sonrasında main fonksiyonu içerisinde verilerin bulunduğu .txt dosyasından verileri çektim ve diziye verileri ekledim.
3. Sıralama algoritmalarının süresini ölçmek için ctime kütüphanesinin sayaç fonksiyonunu sıralamaya başlamadan önce başlattım ve sonrasında bitirdim.
4. Bütün algoritmalar için sıraladıktan sonra en hızlı Merge sort da çıktı alırken, en yavaş Selection sort da çıktı aldım.

```
Selection Sort zaman: 0.000601 saniye  
Insertion Sort zaman: 0.000342 saniye  
Merge Sort zaman: 0.000251 saniye
```

5. Yukarıda da aldığım çıktı gözükmemekte.
6. Merge sort için ortalama durum $O(n \log n)$ iken, Insertion ve Selection sort için $O(n^2)$ 'dir. Aldığım sonuç ile birlikte de bu durumu kanıtlamış oldum.