

## Soru 1:

The screenshot shows a C IDE interface with two tabs: 'soru1.c' and 'soru2.c'. The 'soru1.c' tab contains the following code:

```

1 #include<stdio.h>
2
3 typedef struct {
4     char ad[20];
5     char soyad[20];
6     int not;
7 } student;
8
9 int main() {
10     FILE *file = fopen("students.txt", "w+");
11     student std;
12     int sayac = 0;
13
14     if (file == NULL) {
15         printf("Dosya Acilmadi!\n");
16         return 1;
17     }
18
19     while (sayac < 5) {
20         printf("Ogrencinin adini giriniz: ");
21         scanf("%s", std.ad);
22
23         printf("Ogrencinin soyadini giriniz: ");
24         scanf("%s", std.soyad);
25
26         printf("Ogrencinin notunu giriniz: ");
27         scanf("%d", &std.not);
28
29         //dosyaya yazma islemi
30         fprintf(file, "%s\t%s\t%d\n", std.ad, std.soyad, std.not);
31
32         sayac++;
33     }
34
35     fclose(file);
36     return 0;
37 }
38 */
39
40 rewind(file);
41 printf("\nDosyadaki Ogrenci Verileri");

```

The terminal window shows the output of the program, which reads five student records from the console and writes them to a file named 'students.txt'. The records are then read back and printed to the console.

```

Ogrencinin adini giriniz: ferda
Ogrencinin soyadini giriniz: yargın
Ogrencinin notunu giriniz: 90
Ogrencinin adini giriniz: leyla
Ogrencinin soyadini giriniz: altın
Ogrencinin notunu giriniz: 98
Ogrencinin adini giriniz: zeliha
Ogrencinin soyadini giriniz: yıldırım
Ogrencinin notunu giriniz: 99
Ogrencinin adini giriniz: salih
Ogrencinin soyadini giriniz: erdim
Ogrencinin notunu giriniz: 100
Ogrencinin adini giriniz: turkan
Ogrencinin soyadini giriniz: kepır
Ogrencinin notunu giriniz: 100

Process returned 0 (0x0)   execution time : 55.869 s
Press any key to continue.

```

## Soru 2:

The screenshot shows a C IDE interface with two tabs: 'kelimeler.txt' and 'ters\_kelimeler.txt'. The 'kelimeler.txt' tab contains the following code:

```

Dosya Düzenle Görünüm
kelebek
kus
hur
omur
zaman

```

The 'ters\_kelimeler.txt' tab contains the following code:

```

Kac kelime gireceksiniz? 5
Lutfen kelimeleri girin:
1. kelime: kelebek
2. kelime: kus
3. kelime: hur
4. kelime: omur
5. kelime: zaman

Ters cevrilen kelimeler:
kebelek
suk
ruh
rumo
namaz

Kelimeler 'kelimeler.txt' dosyasina, ters halleri ise 'ters_kelimeler.txt' dosyasina yazildi.

Process returned 0 (0x0)   execution time : 18.450 s
Press any key to continue.

```

The terminal window shows the output of the program, which reads words from 'kelimeler.txt', prints them back, and then writes their reverses to 'ters\_kelimeler.txt'.

## Soru 2 Kod

```

there ✘ soru2.c ✘
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 void terstenKelime(char *kelime) {
6     int uzunluk = strlen(kelime);
7     for (int i = 0; i < uzunluk / 2; i++) {
8         char temp = kelime[i];
9         kelime[i] = kelime[uzunluk - i - 1];
10        kelime[uzunluk - i - 1] = temp;
11    }
12}
13
14 int main() {
15     int kelimeSayisi;
16     char *kelime;
17     FILE *girdiDosyasi, *ciktiDosyasi;
18
19     printf("Kac kelime gireceksiniz? ");
20     scanf("%d", &kelimeSayisi);
21     getchar(); // Tamponu temizle
22
23     // Giris dosyasini yazma modunda ac
24     girdiDosyasi = fopen("kelimeler.txt", "w");
25     if (girdiDosyasi == NULL) {
26         printf("HATA: 'kelimeler.txt' dosyasi olusturulamadi.\n");
27         return 1;
28     }
29
30     printf("Lutfen kelimeleri girin:\n");
31     for (int i = 0; i < kelimeSayisi; i++) {
32         printf("%d. kelime: ", i + 1);
33
34         kelime = (char *)malloc(100 * sizeof(char));
35         if (kelime == NULL) {
36             printf("HATA: Beltek tahsis edilemedi.\n");
37             fclose(girdiDosyasi);
38             return 1;
39         }
40
41         fgets(kelime, 100, stdin);
42         kelime[strcspn(kelime, "\n")] = '\0';
43         fprintf(girdiDosyasi, "%s\n", kelime);
44
45         free(kelime);
46     }
47
48     fclose(girdiDosyasi);
49
50     // Dosyaki okuma modunda ac
51     girdiDosyasi = fopen("kelimeler.txt", "r");
52     if (girdiDosyasi == NULL) {
53         printf("HATA: 'kelimeler.txt' dosyasi acilamadi.\n");
54         return 1;
55     }
56
57     ciktiDosyasi = fopen("ters_kelimeler.txt", "w");
58     if (ciktiDosyasi == NULL) {
59         printf("HATA: 'ters_kelimeler.txt' dosyasi olusturulamadi.\n");
60         fclose(girdiDosyasi);
61         return 1;
62     }
63
64     printf("\nTers cevriken kelimeler:\n");
65     while (1) {
66         kelime = (char *)malloc(100 * sizeof(char));
67         if (kelime == NULL) {
68             printf("Beltek tahsis edilemedi.\n");
69             fclose(girdiDosyasi);
70             fclose(ciktiDosyasi);
71             return 1;
72         }
73
74         if (fgets(kelime, 100, girdiDosyasi) == NULL) {
75             free(kelime);
76             break;
77         }
78
79         kelime[strcspn(kelime, "\n")] = '\0';
80         terstenKelime(kelime);
81         printf("%s\n", kelime);
82         fprintf(ciktiDosyasi, "%s\n", kelime);
83
84         free(kelime);
85     }
86
87     fclose(girdiDosyasi);
88     fclose(ciktiDosyasi);
89
90     printf("\nKelimeler 'kelimeler.txt' dosyasina, ters halleri ise 'ters_kelimeler.txt' dosyasina yaziildi.\n");
91     return 0;
92 }

```