



2405000201 ALGORİTMALAR VE PROGRAMLAMA II
2024 – 2025 Bahar Dönemi
ARA SINAV

Tarih: 09.04.2025

Sınav Süresi: 60 dk

Öğrenci No:	Adı-Soyadı:	Şube No:
Bölümü:	Öğretim Elemanı:	İmza:

Değerlendirme:

Soru 1	Soru 2	Soru 3	Toplam
a) b)	a) b)	a) b)	

SORULAR VE ÇÖZÜMLER

1. a) Aşağıdaki C program kodunda pointer (işaretçi) yöntemi kullanılarak yapılan işlemleri analiz ediniz ve hangi değerlerin ekrana yazdırılacağını belirleyerek ilgili kutucuğa ekran çıktısını yazınız (15 p).

```
#include <stdio.h>

void print_value(int *ptr, int offset) {
    printf("%d ", *(ptr + offset));
}

int main() {
    int arr[] = {10, 20, 30, 40, 50};
    int *p = arr;

    p++;
    print_value(p, 2);

    p = arr;
    print_value(p, 4);

    p++;
    print_value(p, -1);

    p += 2;
    print_value(p, 0);

    p--;
    print_value(p, 1);

    return 0;
}
```

Ekran Çıktısı:

40 50 10 40 40 → her bir çıktı 3 p, 3x5=15 p

b) Aşağıdaki C kodunda boşluklar bulunmaktadır. Boşlukları uygun şekilde doldurarak programın doğru şekilde çalışmasını sağlayınız. Ekran çıktısını ilgili kutucuğa yazınız (20 p).

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

int string_length(char *str) {
    if (str[0] == '\0') { → 3 p
        return 0;
    }
    return 1 + string_length(str + 1); → 3 p
}

int main() {
    char str1[100], str2[100], result[200];

    strcpy(str1, "Recursive");
    strcpy(str2, "Functions");
    strcat(str1, str2); → 3 p + 3 p

    int comparison = strcmp(str1, str2); → 3 p + 3 p

    if (comparison == 0) {
        printf("The strings are equal.\n");
    } else if (comparison < 0) { → 3 p
        printf("str1 is less than str2.\n");
    } else {
        printf("str1 is greater than str2.\n");
    }

    printf("Concatenated result: %s\n", str1);

    int length = string_length(str1); → 3 p
    printf("The length of str1 is: %d\n", length);

    return 0;
}
```

Ekran Çıktısı:

str1 is greater than str2. → 2 p

Concatenated result: RecursiveFunctions → 2 p

The length of str1 is: 18 → 2 p

2. a) Aşağıdaki C program kodunda rekürsif/özyinelemeli yöntemi kullanılarak yapılan işlemleri analiz ediniz ve hangi değerlerin ekrana yazdırılacağını belirleyerek ilgili kutucuğa ekran çıktısını yazınız (15 p).

```
#include <stdio.h>

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

int main() {
    for (int i = 0; i < 6; i++) {
        printf("%d ", callFunction (i));
    }
    return 0;
}
```

Ekran Çıktısı:

0 1 1 2 3 5 → 2 puan x 6 = 12 puan. Doğru sıralama = 3 Puan. Toplam = 15 Puan

- b) Aşağıdaki C kodunda bir karakterin dizideki tekrar sayısı için bir fonksiyon kullanılmaktadır. Eksik fonksiyonu tamamlayıp programın doğru şekilde çalışmasını sağlayınız. Ekran çıktısını ilgili kutucuğa yazınız (20 p).

```
#include <stdio.h>

int countOccurrences(char *str, char ch) {
    int count = 0;
    while (*str != '\0') {
        if (*str == ch) {
            count++;
        }
        str++;
    }
    return count;
}

int main() {
    char str[] = "gobekli tepe";
    char ch = 'e';
    printf("%c' nin tekrar sayısı: %d\n", ch, countOccurrences(str, ch));
    return 0;
}
```

Ekran Çıktısı:

e nin tekrar sayısı: 3 → 3 Puan

Doğru çalışan ve karakter sayısını doğru bulan kod = 17 Puan, not: Soru açık uçludur, çözüm dışındaki alternatif yöntemler de çalışıyorsa kabul edilecektir.

3. a) Aşağıdaki tamsayı dizisinin elemanlarını **Quicksort** (Hızlı Sıralama) algoritmasına göre adım adım ilerleyerek sıralamanız istenmektedir. Her adımda alt dizilerde **pivot** olarak **ilk elemanı** seçmeniz gerekmektedir (15p).

Pivot

Sıralanacak Başlangıç Dizisi:

21	18	34	13	25	19	72	5	47
----	----	----	----	----	----	----	---	----

1. Adım: $18 \rightarrow 13 \rightarrow 19 \rightarrow 5 \leftarrow 21 \rightarrow 34 \rightarrow 25 \rightarrow 72 \rightarrow 47$ (Küçükler sola, Büyükler sağa)
2. Adım: $13 \rightarrow 5 \leftarrow 18 \rightarrow 19$ (Tek eleman), $25 \leftarrow 34 \rightarrow 72 \rightarrow 47$ (Tek eleman)
3. Adım: $5 \leftarrow 13$ (Tek eleman), $47 \leftarrow 72$ (Tek eleman)
4. Adım: Tek elemana düşen alt dizide pivot belirlemeye gerek yok
5. Adım:

Sıralanmış Dizi:

5	13	18	19	21	25	34	47	72
---	----	----	----	----	----	----	----	----

Sıralanmış Dizi:

5	13	18	19	21	25	34	47	72
---	----	----	----	----	----	----	----	----

- b) Yukarıda **Quicksort** (Hızlı Sıralama) algoritmasına göre küçükten büyüğe doğru sıraladığınız dizinin içinde **Binary Search** (İkili Arama) algoritmasını kullanarak adım adım ilerleyip 63 sayısını aramanız istenmektedir (15p).

- Sıralanmış dizinin orta elemanı ile aranan sayıyı karşıla ($63 > 21$)
- Aranan sayı orta elemenden büyük o zaman sağ tarafta ara

25	34	47	72
----	----	----	----
- Aranan sayı ile orta elemanı karşıla (34'te olabilir, 47'de olabilir fark etmez.) ($63 > 47$)
- Aranan sayı orta elemenden büyük sağ tarafta ara

72

- Sağ tarafta tek eleman kaldı (72), karşıla ($63 \neq 72$)
- Aranan eleman bulunamadı.