

## Rekürsif Sorular

1. **fun** fonksiyonu main fonksiyonundan **n=0, a=5 ve b=7** değeri için çağırıldığındaki ekran çıktısını yazın.  
Write screen output, when **fun** is called in main function with **n=0, a=5 and b=7**.

<pre>void fun(int n, int a, int b) {     if (n &gt;= 10)         return;     fun(n + 2, a, b + n);     printf("%d %d %d\n", n, a, b); }</pre>	<b>OUTPUT</b>
---	---------------

2. Dizinin adresini ve eleman sayısını parametre olarak alan ve dizinin içinde kaç tane **tek** sayı olduğunu döndüren recursive bir fonksiyon yazınız. (**Global değişken kullanmayınız**). Write a recursive function that takes the address of the array and the number of elements as a parameter and returns the number of **odd** numbers in the array. (**Do not use global variable**)

3. What is the output of the following C program? Aşağıdaki C programının çıktısı nedir?

```
#include <stdio.h>

int fun(int n)
{
    if (n == 4)
        return n;
    else return 2*fun(n+1);
}

int main()
{
    printf("%d ", fun(2));
    return 0;
}
```

4. What is the output of the following C program? Aşağıdaki C programının çıktısı nedir?

```
int toplam(int n){
    if (n == 1)
        return n;
    else
        return (n + toplam(n - 1));
}

void main(void){
    int sayi = 6;
    printf("Sonuc = %d", toplam(sayi));
}
```

5. In recursion, the condition for which the function will stop calling itself is \_\_\_\_\_  
Özyinelemede, işlevin kendini çağırmaı durduracağı koşul \_\_\_\_\_ ile ifade edilir.
- a) Best case (En iyi durum)
  - b) Worst case (En kötü durum)
  - c) Base case (Temel durum)
  - d) There is no such condition (Böyle bir durum yoktur)

6. What does the following recursive code do? Aşağıdaki özyinelemeli kod hangi işlemi yapar?

```
void my_recursive_function(int n){
    if(n == 0)
        return;
    printf("%d ",n);
    my_recursive_function(n-1);
}
void main(){
    my_recursive_function(7);
}
```

7. What does the following function print for n = 25? n = 25 için aşağıdaki rekürsif fonksiyon ekrana ne yazdırır?

```
void fun(int n){
    if (n == 0)
        return;

    fun(n/2);
    printf("%d", n%2);
}
```

8. You have to find the sum of digits of a number given that the number is always greater than 0. Which of the following base cases can replace the base case for the below code?

Sayının her zaman 0'dan büyük olduğu durumlar için bir sayının rakamlarının toplamını bulmak istiyoruz. Aşağıdaki temel durumlardan hangisi kod içindeki temel durum ile değiştirilmelidir?

```
int recursive_sum_of_digits(int n){
    if(n == 0)
        return 0;
    return n % 10 + recursive_sum_of_digits(n/10);
}
void main(){
    int n = 1201;
    int ans = recursive_sum_of_digits(n);
    printf("%d",ans);
}
```

- A. if(n == 0) return 1
- B. if(n == 1) return 0
- C. if(n == 1) return 1
- D. if(n == 0) return 0
- E. None of them (Hiçbiri)

9. If the following code is executed, how many "a" will be written on the screen.  
Aşağıdaki verilen kod çalıştırılırsa ekran kaç adet "a" yazar.

```
#include<stdio.h>
char f1(int b);
main()
{
    char a;
    a=f1(10);
    printf("%c",a);
}
char f1(int b)
{
    printf("a");
    if(b<=8)
        return 'a';
    else
    {
        printf("a");
        f1(b--);
    }
}
```

- A. 6  
B. 12  
C. 10  
D. 2  
E. Sınırsız sayıda(Infinitive number)