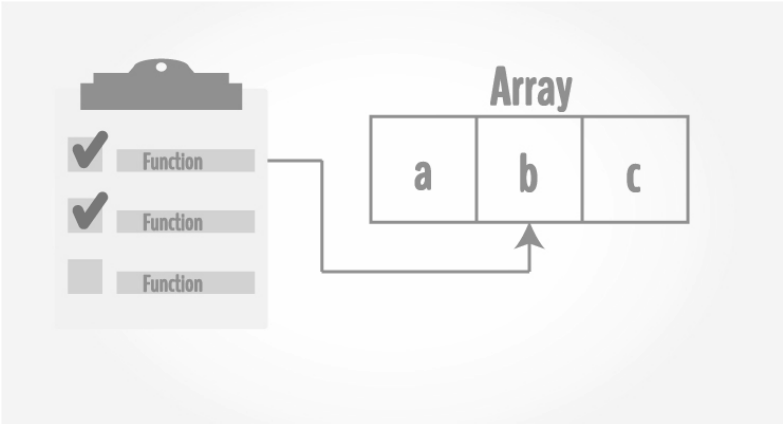
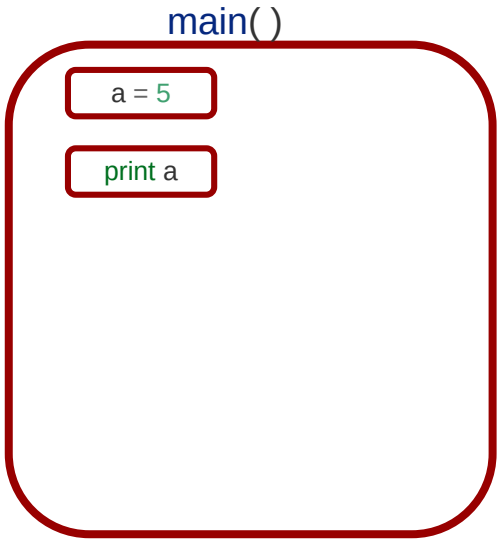
# Diziler




Suhap SAHIN
Onur GÖK
Fidan Kaya Gülağız

# Fonksiyon & Dizi

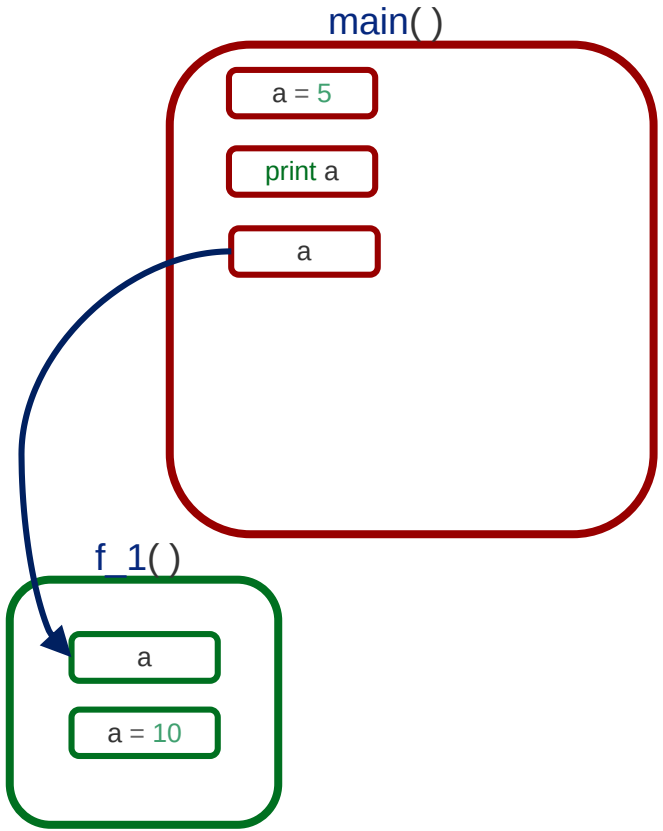



```
int main() {
        int a = 5;
        printf("a: %d\n", a)

        return 0;
}
```

# Fonksiyon & Dizi

```
void f_1(int a) {

}
```

```
int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);

        return 0;
}
```

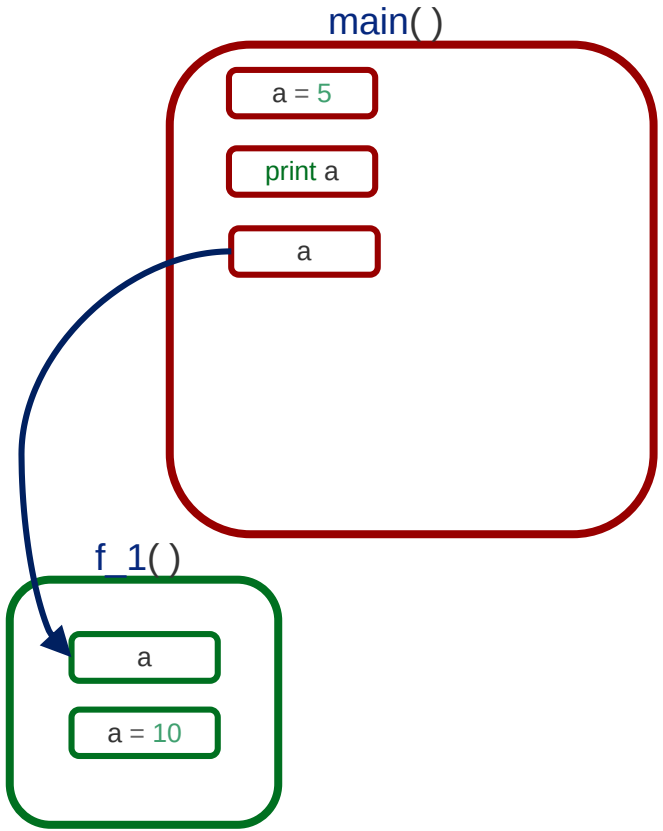


# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}
```

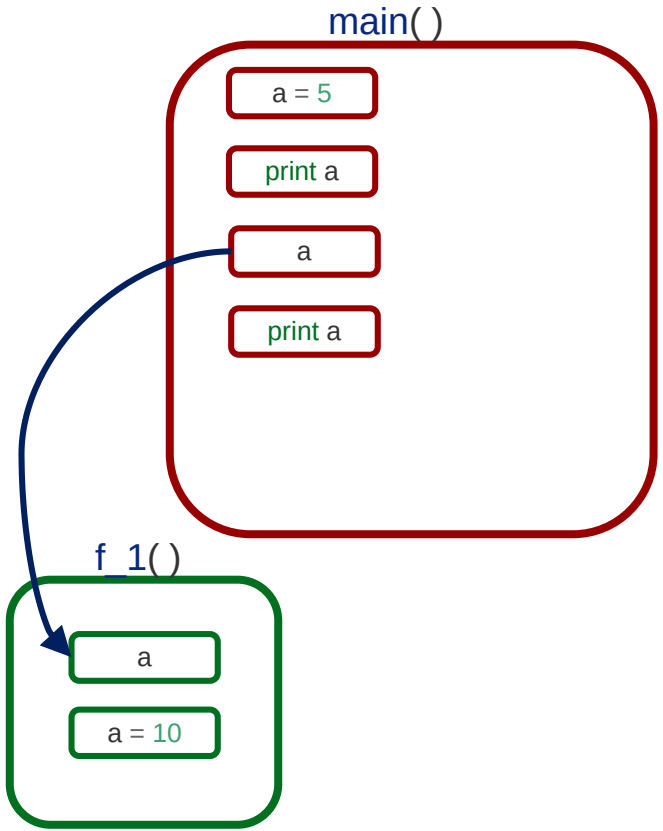
```
int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
```

```
        return 0;
}
```



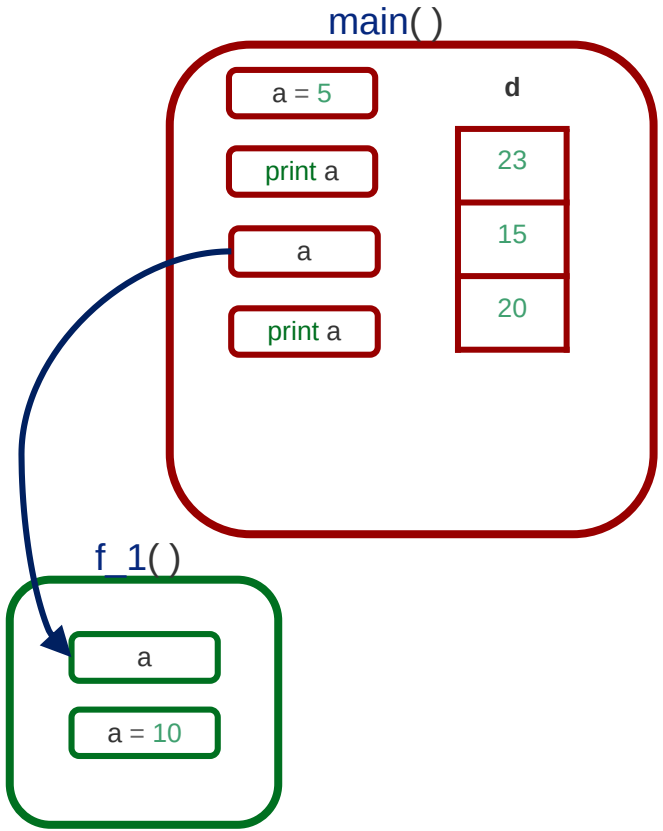
# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}


int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);


        return 0;
}
```



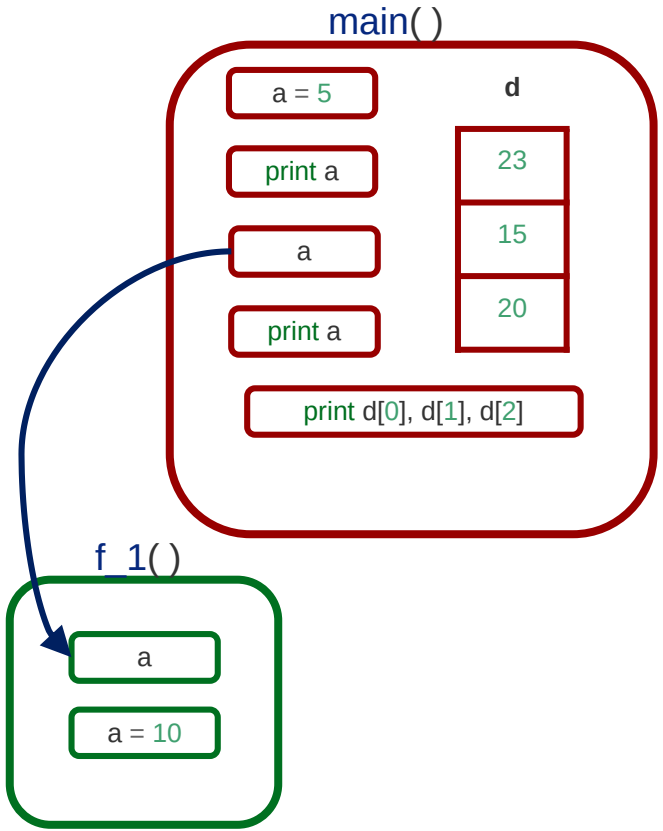
# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}


int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);
        int d[3] = {11, 22, 33};


        return 0;
}
```



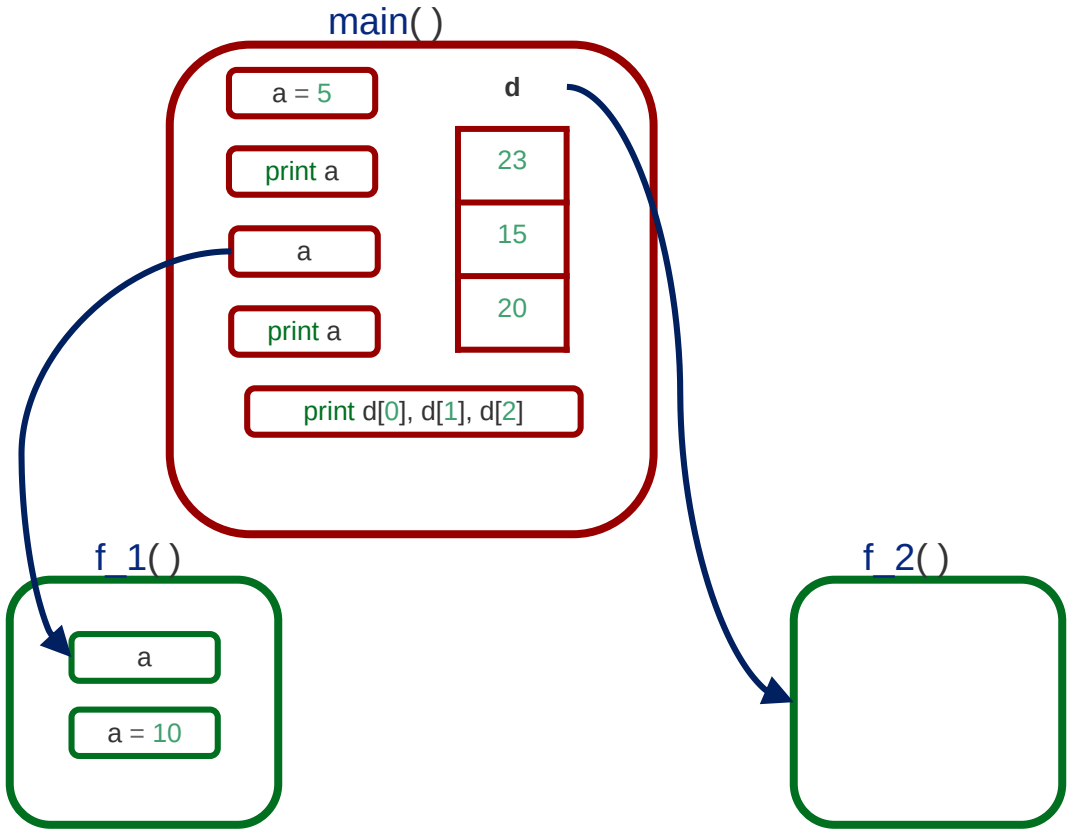
# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}

int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);
        int d[3] = {11, 22, 33};
        printf("d: %d %d %d\n", d[0], d[1],
d[2]);

        return 0;
}
```



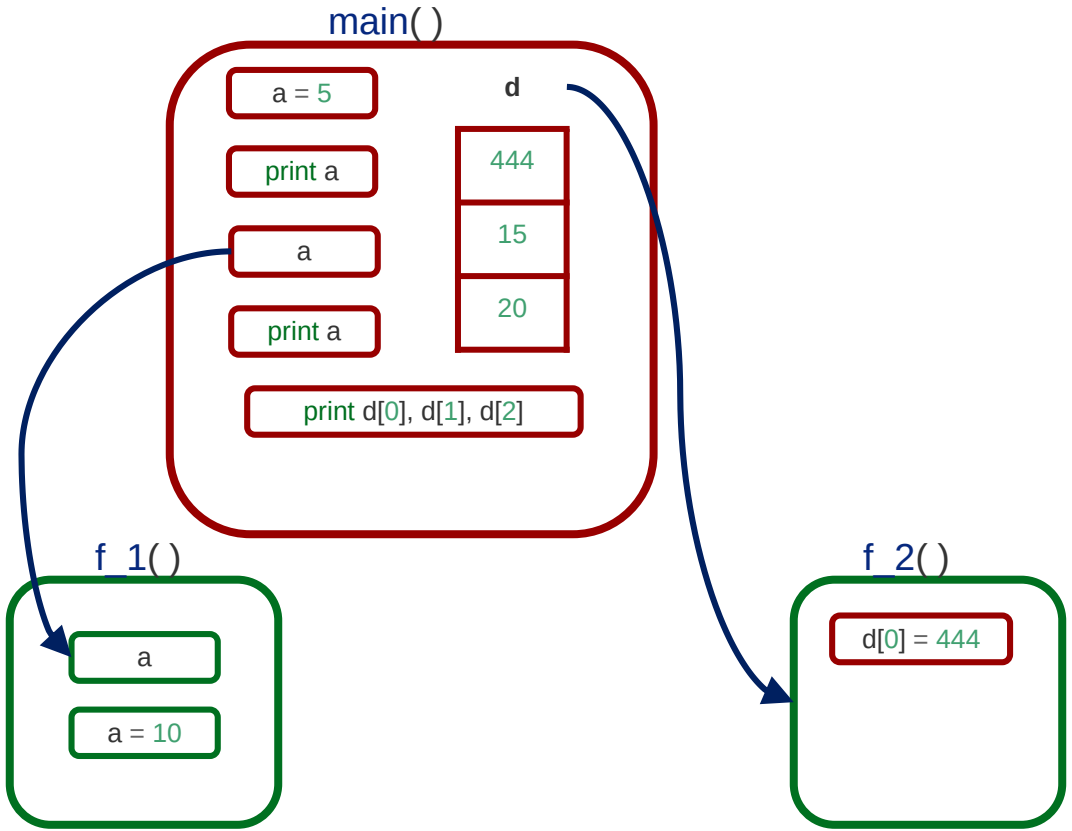
# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}
void f_2(int d[]) {




}
int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);
        int d[3] = {11, 22, 33};
        printf("d: %d %d %d\n", d[0], d[1],
d[2]);
        f_2(d);


        return 0;
}
```



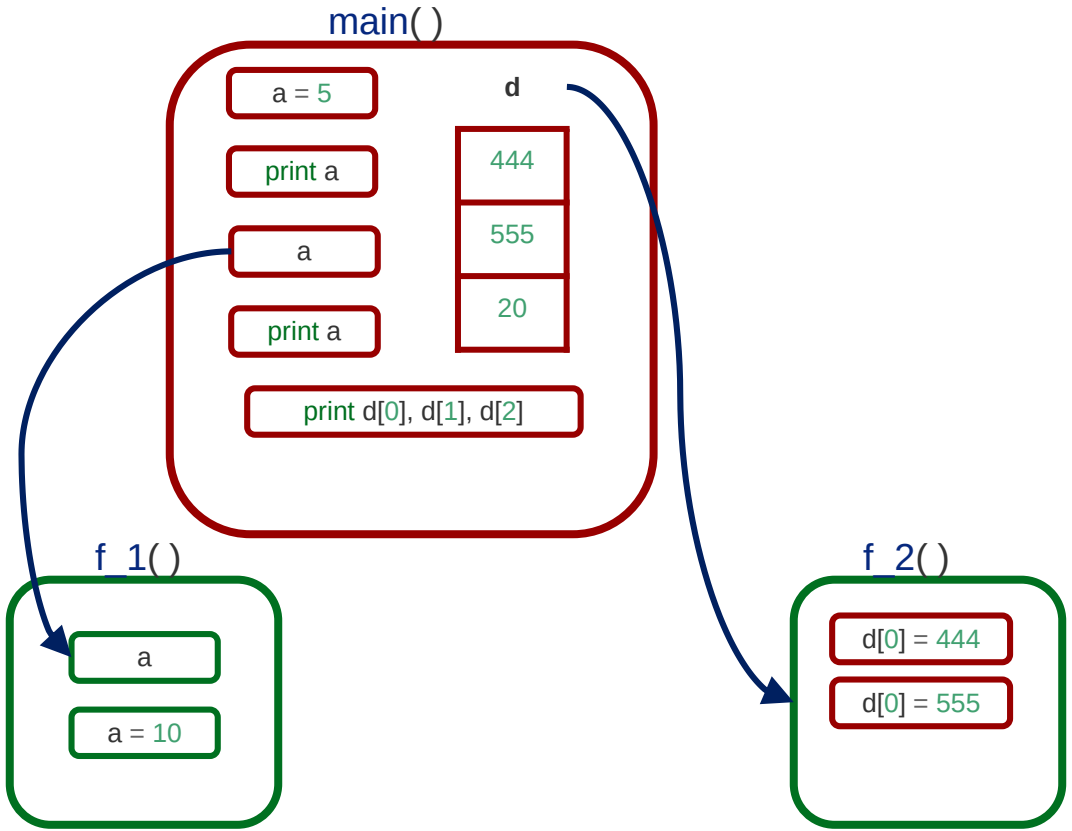
# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}
void f_2(int d[]) {
        d[0] = 444;



}
int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);
        int d[3] = {11, 22, 33};
        printf("d: %d %d %d\n", d[0], d[1],
d[2]);
        f_2(d);



        return 0;
}
```



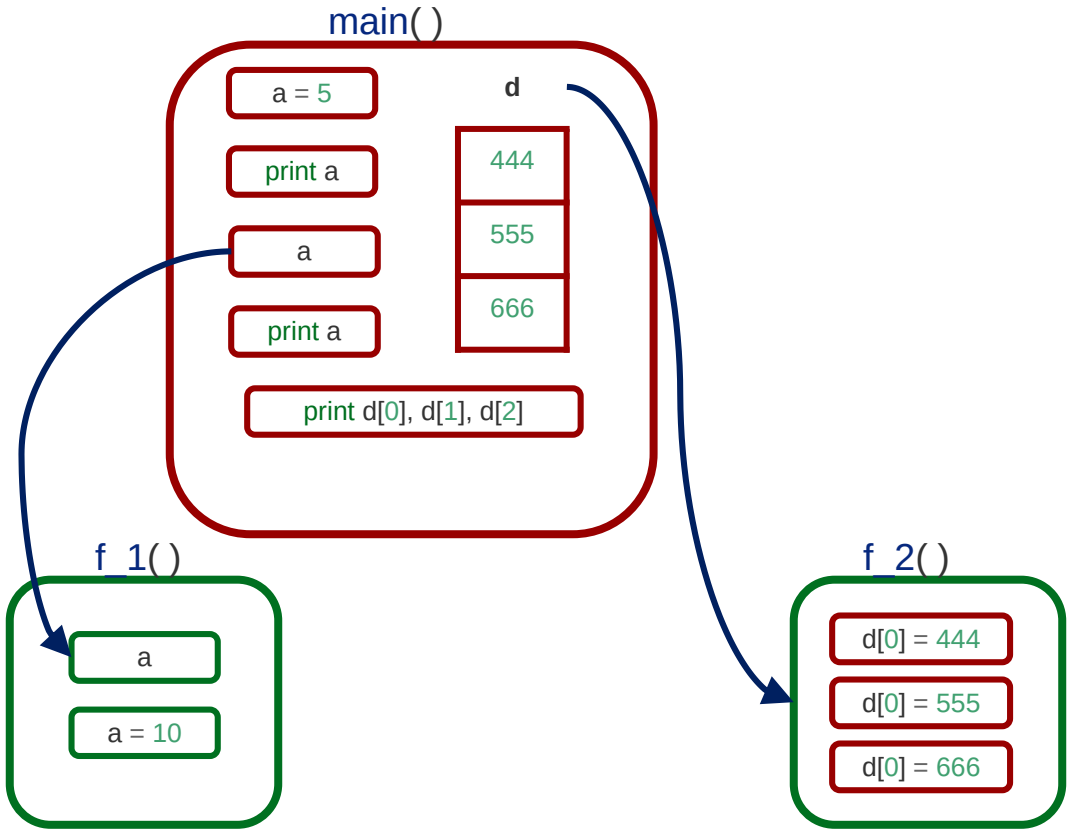
# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}
void f_2(int d[]) {
        d[0] = 444;
        d[1] = 555;

}
int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);
        int d[3] = {11, 22, 33};
        printf("d: %d %d %d\n", d[0], d[1],
d[2]);
        f_2(d);


        return 0;
}
```



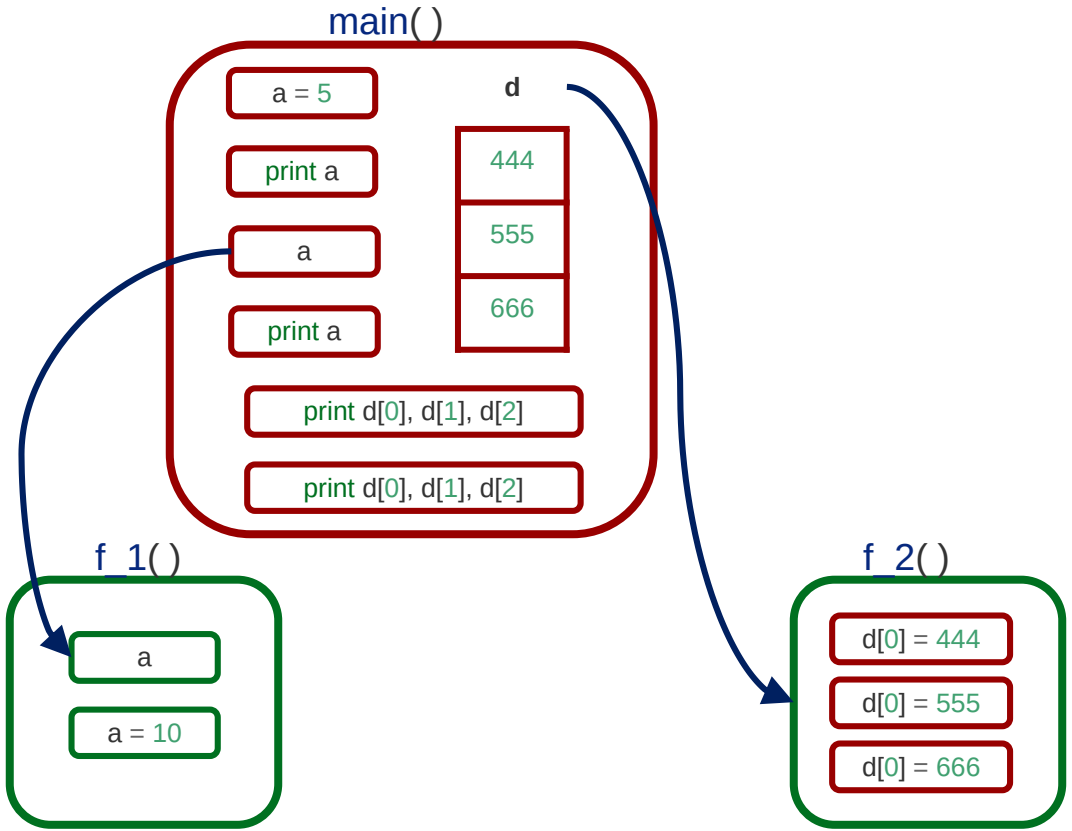
# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}
void f_2(int d[]) {
        d[0] = 444;
        d[1] = 555;
        d[2] = 666;
}
int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);
        int d[3] = {11, 22, 33};
        printf("d: %d %d %d\n", d[0], d[1],
d[2]);
        f_2(d);


        return 0;
}
```

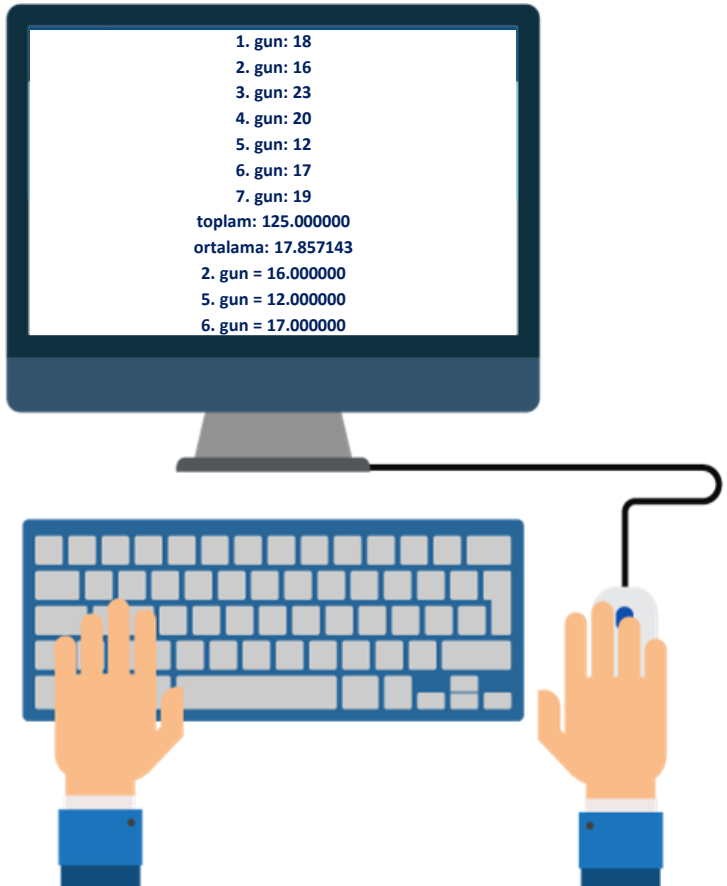


# Fonksiyon & Dizi

```
void f_1(int a) {
        a = 10;
}
void f_2(int d[]) {
        d[0] = 444;
        d[1] = 555;
        d[2] = 666;
}
int main() {
        int a = 5;
        printf("a: %d\n", a)
        f_1(a);
        printf("a: %d\n", a);
        int d[3] = {11, 22, 33};
        printf("d: %d %d %d\n", d[0], d[1],
d[2]);
        f_2(d);
        printf("d: %d %d %d\n", d[0], d[1],
d[2]);
        return 0;
}
```



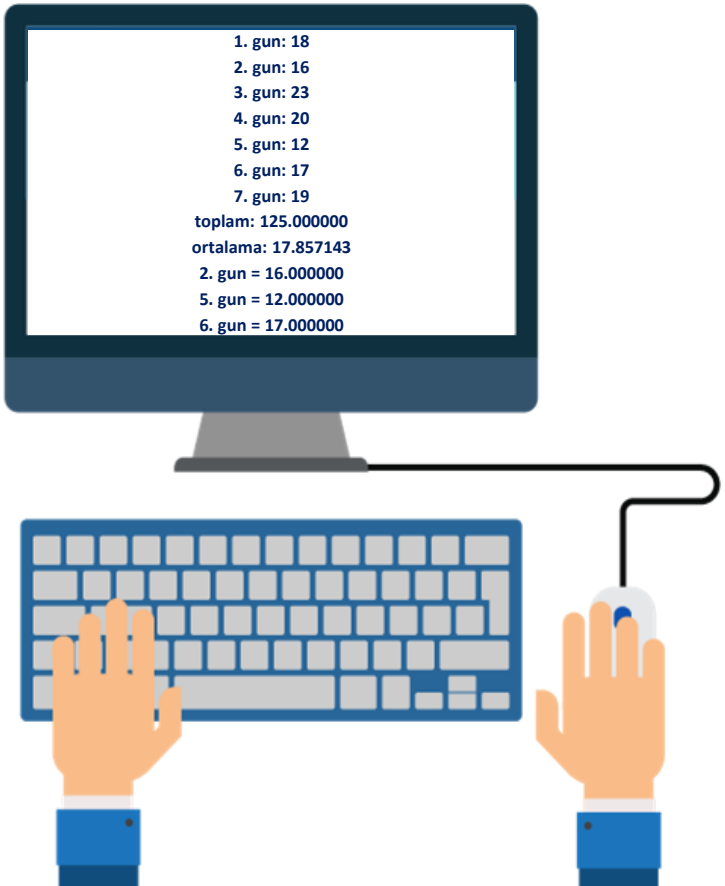
# Ortalama sıcaklık



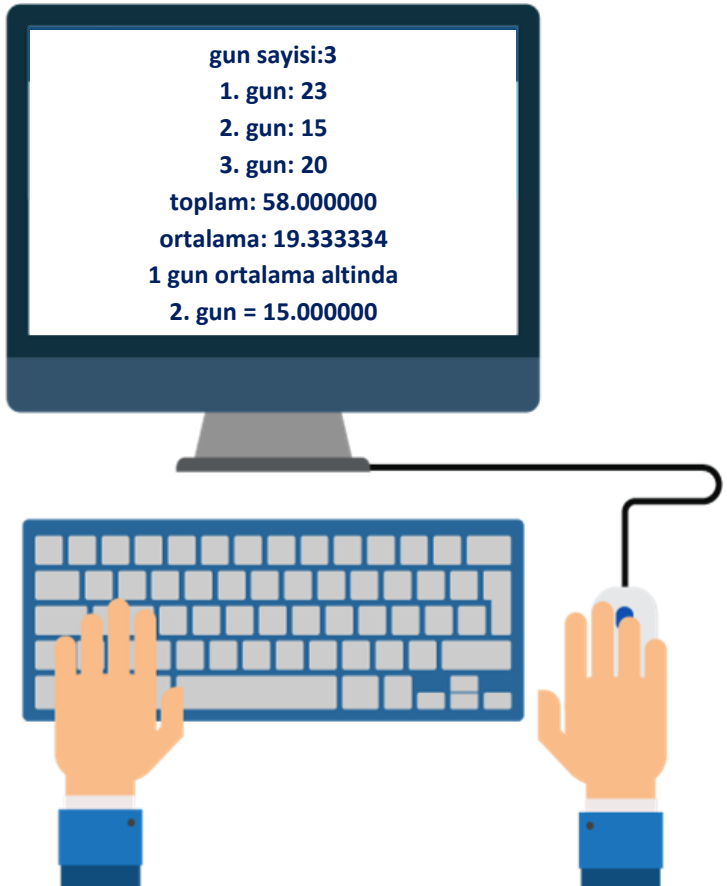

| index | icerik | adress | |
|---|---|---|---|
| | ... | ... | |
| | | 0F1C | |
| 0 | 18 | 0F20 | ← sicakliklar |
| 1 | 16 | 0F24 | |
| 2 | 23 | 0F28 | |
| 3 | 20 | 0F2C | |
| 4 | 12 | 0F30 | |
| 5 | 17 | 0F34 | |
| 6 | 19 | 0F38 | |
| | ... | ... | |

# Ortalama sıcaklık

```c
#include <stdio.h>
float dizi_toplam(float d[], int eleman_say) {
    int i;
    float toplam = 0;
    for (i = 0 ; i < eleman_say ; i++) {
        toplam += d[i];
    }
                        return toplam;
}
int main() {
    float sicakliklar[7];
    int i;
    for (i = 0 ; i < 7 ; i++) {
        printf("%d. gun: ", i+1);
        float x;
        scanf("%f", &x);
        sicakliklar[i] = x;
    }
    float toplam = dizi_toplam(sicakliklar, 7);
    printf("toplam: %f\n", toplam);
    float ortalama = toplam / 7.0;
    printf("ortalama: %f\n", ortalama);
    for (i = 0 ; i < 7 ; i++) {
        if (sicakliklar[i] < ortalama) {
            printf("%d. gun = %f\n", i+1, sicakliklar[i]);
        }
    }
}
```



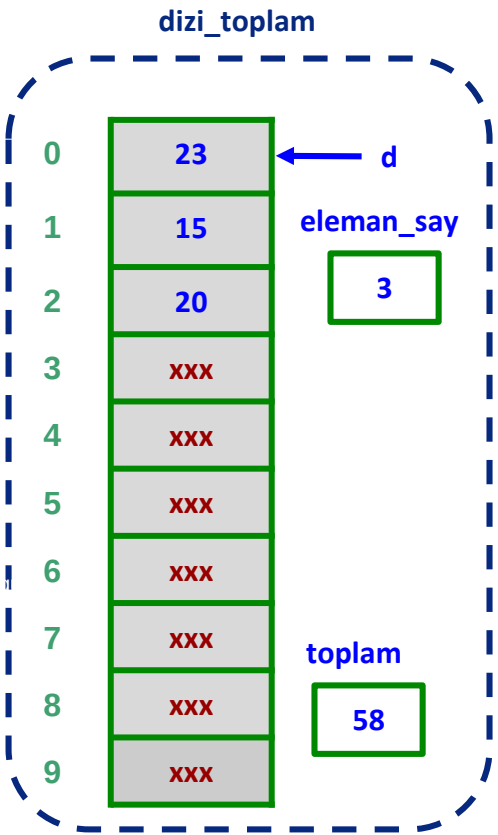
# Ortalama altındaki sıcaklık

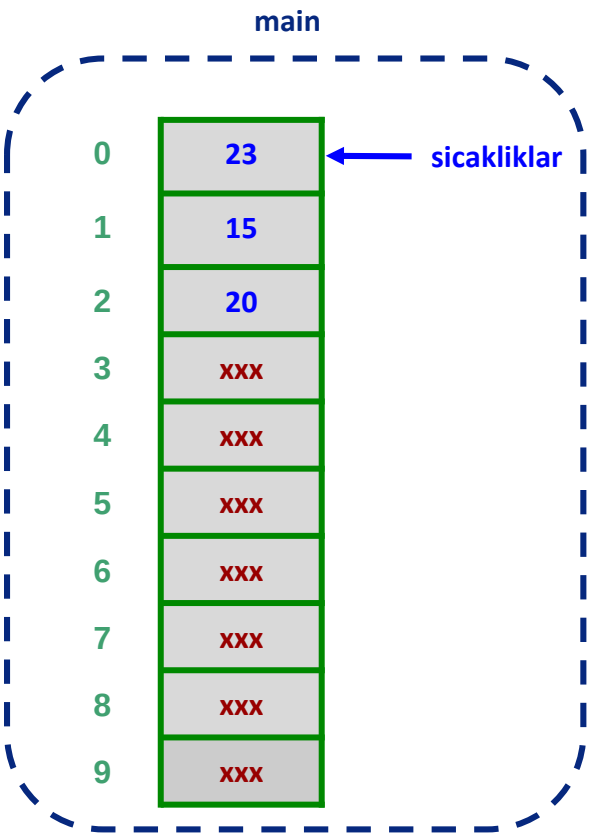



| index | icerik | adress | |
|---|---|---|---|
| 0 | 23 | 0F1C | ← sicakliklar |
| 1 | 15 | 0F20 | |
| 2 | 20 | 0F24 | |
| 3 | xxx | 0F28 | |
| 4 | xxx | 0F2C | |
| 5 | xxx | 0F30 | |
| 6 | xxx | 0F34 | |
| 7 | xxx | 0F38 | |
| 8 | xxx | 0F40 | |
| 9 | xxx | 0F44 | |

# Ortalama altındaki sıcaklık

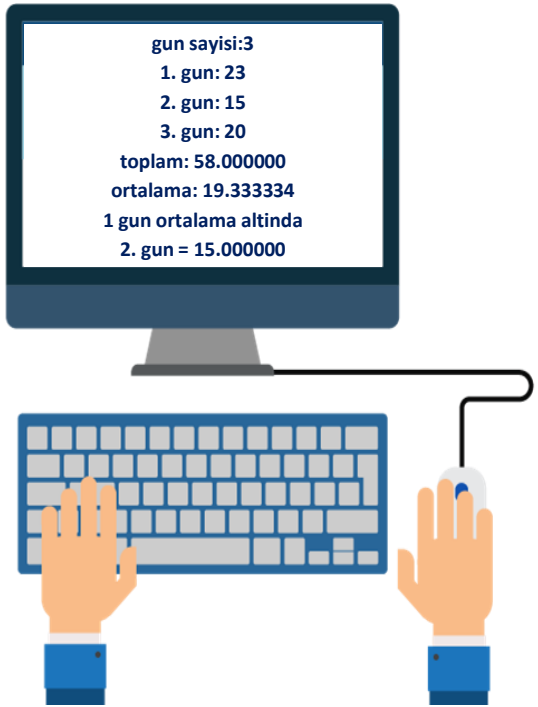
```c
#include <stdio.h>
float dizi_toplam(float d[], int eleman_say);
int main() {
        float sicakliklar[10];
        int N, i;
        printf("gun sayisi:");
        scanf("%d", &N);
        for ( i = 0 ; i < N ; i++) {
                printf("%d. gun: ", i+1);
                scanf("%f", &sicakliklar[i]);
        }
        float toplam = dizi_toplam(sicakliklar, N);
}
float dizi_toplam(float d[], int eleman_say) {
        int i;
        float toplam = 0;
        for (i = 0 ; i < eleman_say ; i++)
                toplam += d[i];
        return toplam;
}
```



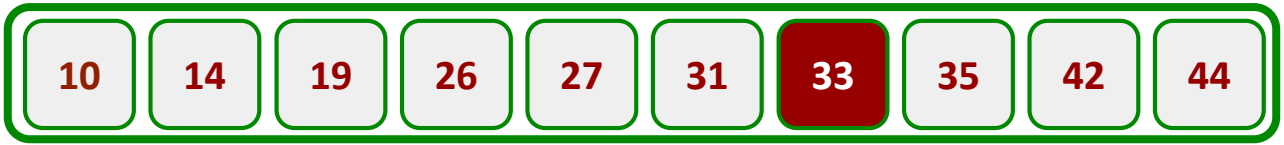



# Ortalama altındaki sıcaklık

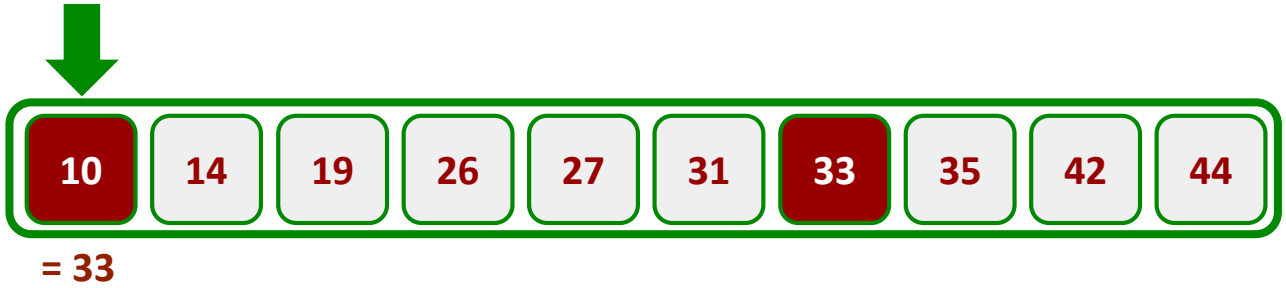
```c
#include <stdio.h>
float dizi_toplam(float d[], int eleman_say);
int main() {
        float sicakliklar[10];
        int N, i;
        printf("gun sayisi:");
        scanf("%d", &N);
        for ( i = 0 ; i < N ; i++) {
                printf("%d. gun: ", i+1);
                scanf("%f", &sicakliklar[i]);
        }
        float toplam = dizi_toplam(sicakliklar, N);
        printf("toplam: %f\n", toplam);
        float ortalama = toplam / (float)N;
        printf("ortalama: %f\n", ortalama);
        int sayac = 0;
        for (i = 0 ; i < N ; i++)
                if (sicakliklar[i] < ortalama)
                        sayac++;
        printf("%d gun ortalama altinda\n", sayac);
        for (i = 0 ; i < N ; i++)
                if (sicakliklar[i] < ortalama)
                        printf("%d. gun =
%f\n", i+1, sicakliklar[i]);
}
float dizi_toplam(float d[], int eleman_say) {
        int i;
        float toplam = 0;
        for (i = 0 ; i < eleman_say ; i++)
                toplam += d[i];
        return toplam;
}
```



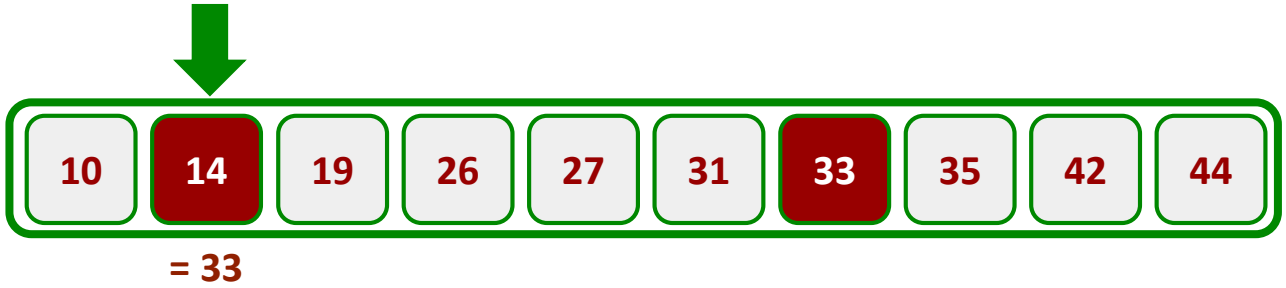
# Dogrusal Arama



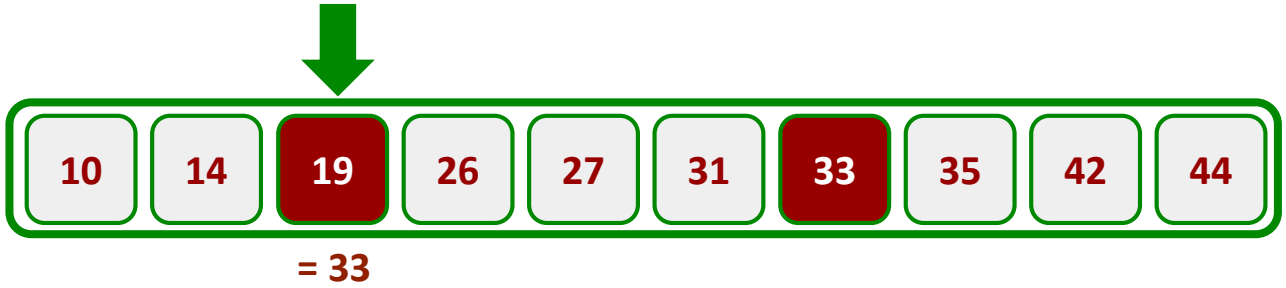
# Dogrusal Arama



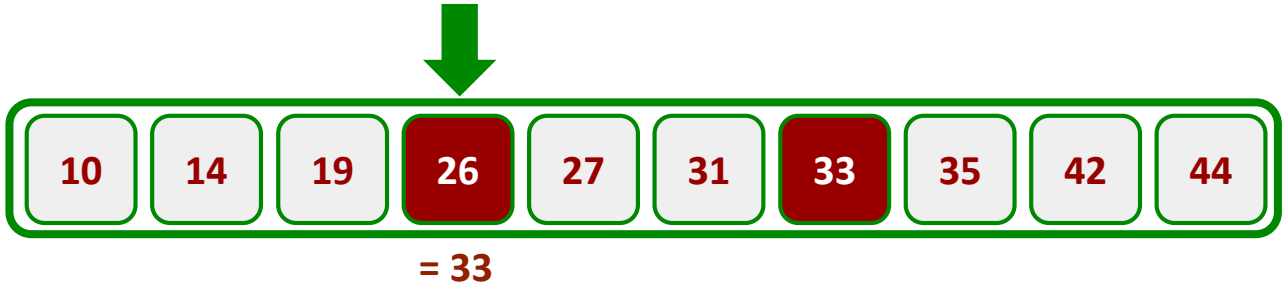
# Dogrusal Arama



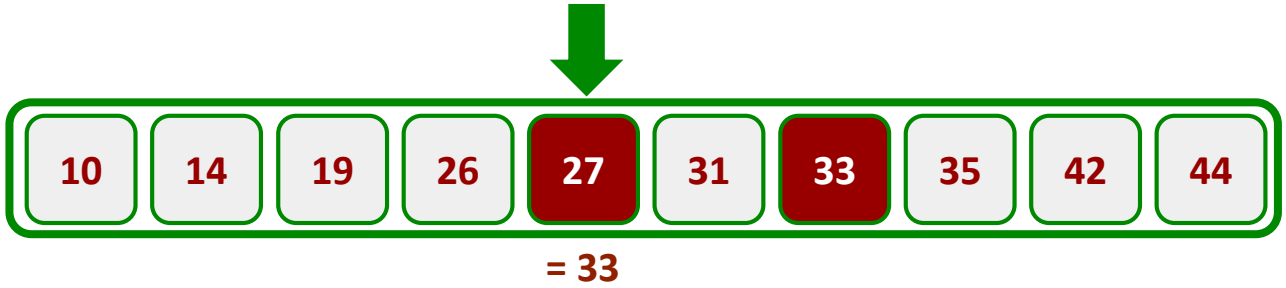
# Dogrusal Arama



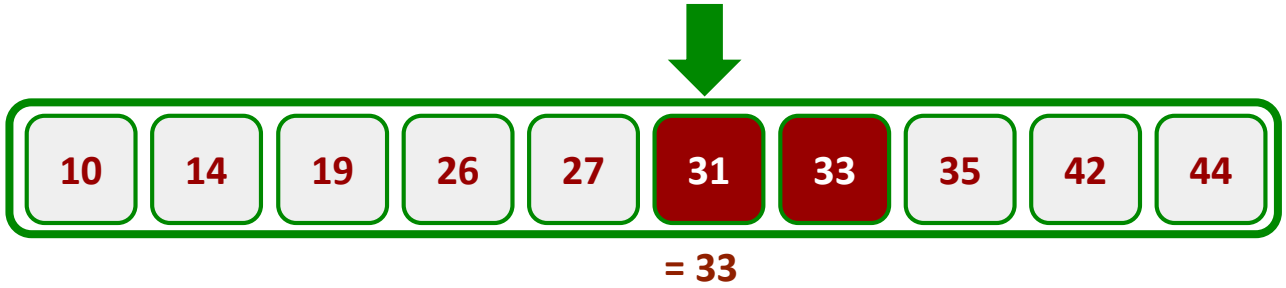
# Dogrusal Arama



# Dogrusal Arama



# Dogrusal Arama



# Dogrusal Arama


# Dogrusal Arama

```c
#include <stdio.h>
int main() {
    int n[10] = {10, 14, 19, 26, 27, 11, 33, 35, 42, 44};
    while (1) {









    }
    return 0;
}
```

# Dogrusal Arama

```c
#include <stdio.h>
int main() {
    int n[10] = {10, 14, 19, 26, 27, 11, 33, 35, 42, 44};
    while (1) {
        int i, aranan;
        printf("aramak istediginiz sayiyi giriniz:");
        scanf("%d", &aranan);



    }
    return 0;
}
```

# Dogrusal Arama

```c
#include <stdio.h>
int main() {
    int n[10] = {10, 14, 19, 26, 27, 11, 33, 35, 42, 44};
    while (1) {
        int i, aranan;
        printf("aramak istediginiz sayiyi giriniz:");
        scanf("%d", &aranan);
        int yer = -1;




    }
    return 0;
}
```

# Dogrusal Arama

```
#include <stdio.h>
int main() {
    int n[10] = {10, 14, 19, 26, 27, 11, 33, 35, 42, 44};
    while (1) {
        int i, aranan;
        printf("aramak istediginiz sayiyi giriniz:");
        scanf("%d", &aranan);
        int yer = -1;
        for ( i = 0 ; i < 10 ; i++ )
            if (n[i] == aranan)
                yer = i;



    }
    return 0;
}
```

# Dogrusal Arama

```c
#include <stdio.h>
int main() {
    int n[10] = {10, 14, 19, 26, 27, 11, 33, 35, 42, 44};
    while (1) {
        int i, aranan;
        printf("aramak istediginiz sayiyi giriniz:");
        scanf("%d", &aranan);
        int yer = -1;
        for ( i = 0 ; i < 10 ; i++ )
            if (n[i] == aranan)
                yer = i;
        if (yer == -1)
            printf("dizide yok\n");
        else
            printf("dizide bulundugu yer: %d\n", yer);
    }
    return 0;
}
```

# *Sorular*