Fonksiyonlar

printf → Çağırıyoruz (call)
scanf → Çağırmak (invoke)
main

<math.h> abs(x)

```
int main(){

    return 0;
}
```

int → return value
return type

() → argüman

abs(x) → argüman

return type name(arg...)
void ...

```
int sayi=10
```

$\boxed{\texttt{kare(sayi);}}$ → 10

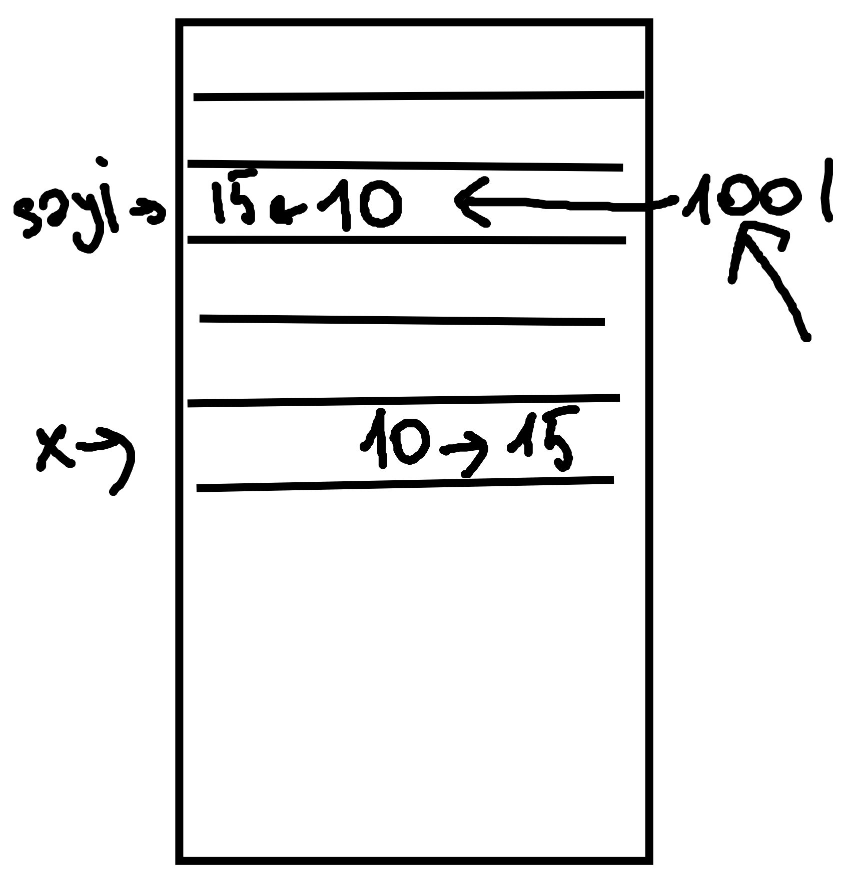
```
int kare(int x){
    int x =10;
    x *= x;
    x=10*10
    return x;   →100
}
```

by value

```
int sayi=10
kare(sayi);
```

```
int kare(int x){
             sayi
    x *= x
    return x;
}
```

by reference



by value

x=10

by reference

x =1001

*x = 10

*x += 5;

```
int main(){

}
```

scope

```
int kare(int x){

}
```

scope

```
for( ; ; ){

}
```

scope

```
int main(){
    int sayi;

}
```

```
int kare(){
    sayi *= sayi;
}
```

$$\begin{tikzcd} & \text{Yerel} \arrow[dl] \arrow[dr] & \\ \text{normal} & & \text{static} \end{tikzcd}$$

Global

herkes erişebilir.

```
int f(){          int f2(){
  int z;            static int y;
}                 }

f(); → z          f2(); → y
f(); → z          f2();
f(); → z          f2();
```

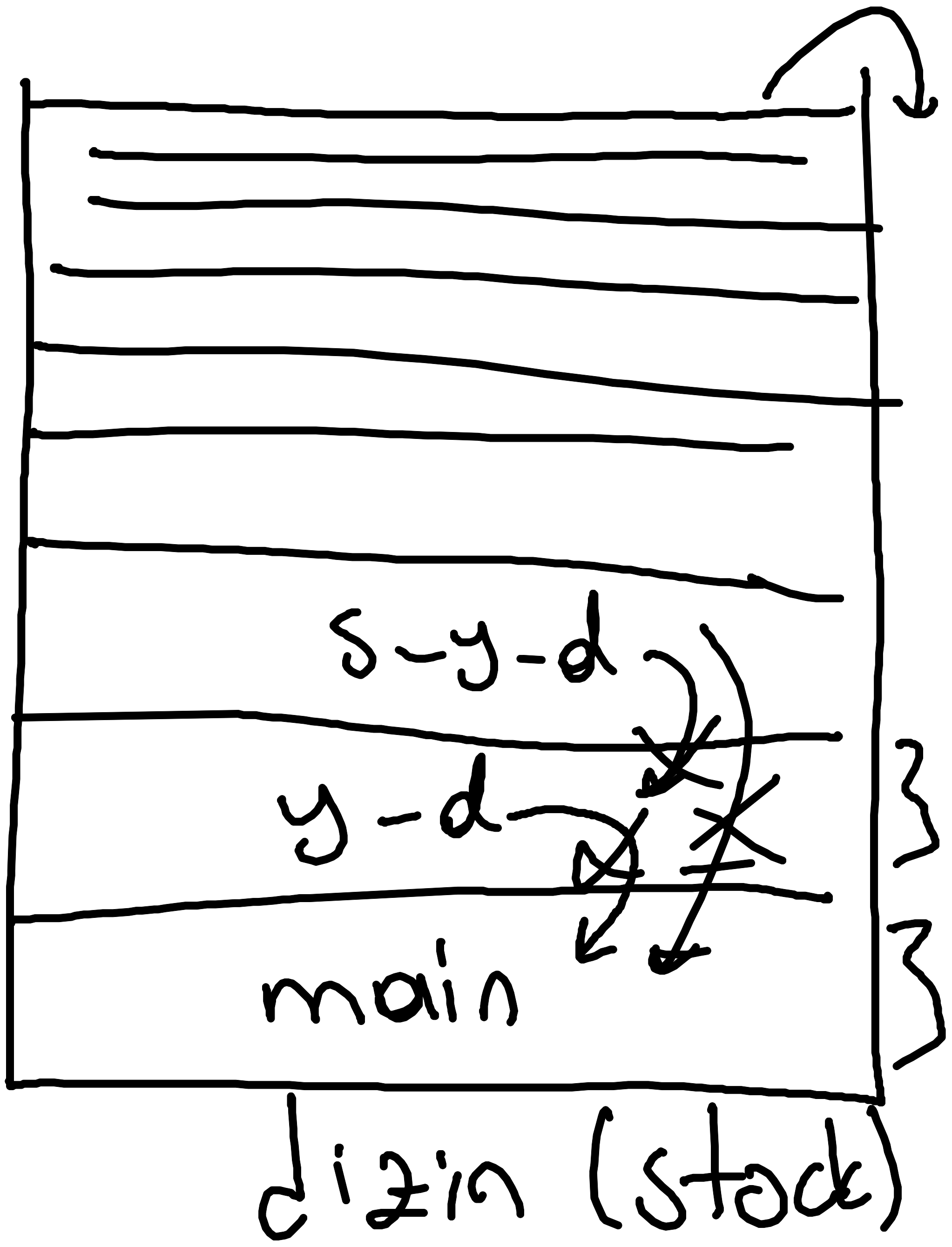
```
global{          main{           {
   x=1 ;           x=10;          x=7;
}                }               }

yerel{           statik{         global{
  x=30 ;           x=50 ;          x-=100;
}                }               }
```

main
y-d
s-y-d
global

x = 1

return 0;



stackoverflow