

İşaretçiler (Pointers): C ve Cpp dilinde değişkenlerin adreslerini tutan yapıdır.

Örnekler:

int *p: p ifadesi bir pointer (işaretçidir) ve sadece adres tutar.

Bu adresin içeriği integer (tam sayı) olsun.

Dikkat: (p) ifadesi pointer, (*p) değil

p = &myvar: p pointerına myvar adlı değişkenin adresini ata komutu

var = *p: p işaretçisinin içeriğini var değişkenine ata komutu.

Örnek 1

int *p;
int myvar = 25; (myvar isimli değişkene 25 ata)

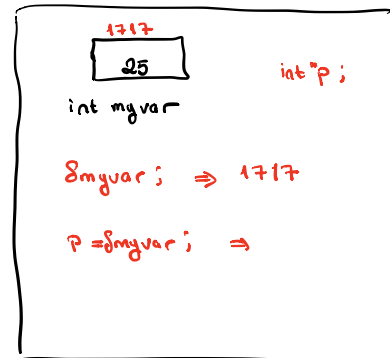
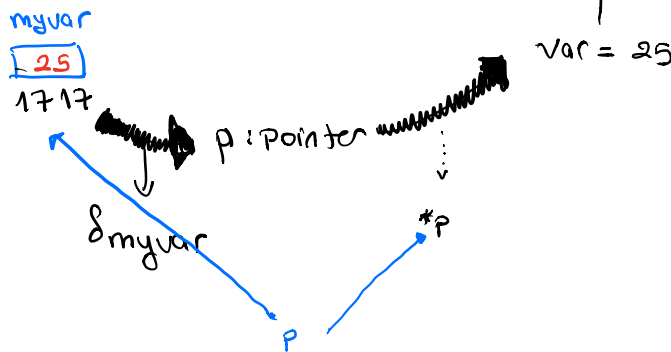
p = &myvar (p pointerına myvar değişkeninin adresini ata)

var = *p (p pointerının içeriğini var değişkenine ata)

printf (" myvar = %d \n", myvar); // myvar = 25

printf (" p = %d \n", p); // p = 1717

printf (" var = %d \n", var); // var = 25

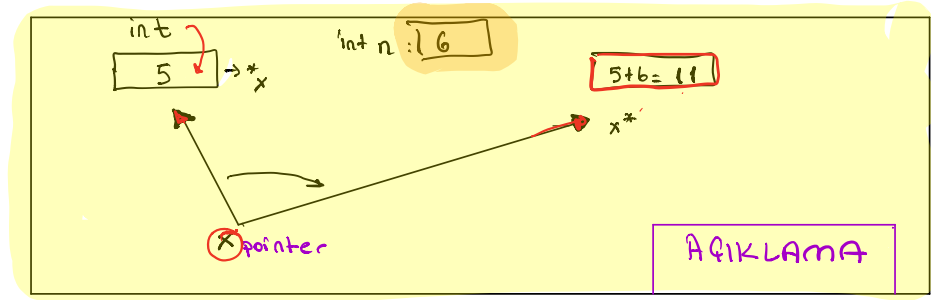


Örnek Program: Değişkenin adresi üzerinde artım yapan bir fonksiyon.

program çıktısı ←

```
#include "stdio.h"

void n_arttır( int *x, int n )
{
    *x = *x + n; → x pointerinin içeriğini n ile toplar ve pointer'ı ortaya gelir.
    printf( " a= %d \n", *x )
}
```



void main ()

{

// Tanımlamalar

int a = 5;

int c = 88;

int *b;

// b tam sayı türünde bir işaretçi

n_arttır(&a, 7); → 12

Pointer kullanımı: *b = 10
n_arttır(b, 7); → 17

char;
scanf("%c", &x); } Visual C'de ekranı tutmak için!
}

DİKKAT: $\text{int}^*b \Rightarrow$ işaretçisi ayarlayalım.

SIRA ÖNEMLİ!

Eğer kod sırasında

$\text{int } c = 2;$

$b = \&c$ (adresini işaretçiye at!)

$c = 71;$ (kod sırasında c değişkeni değişirse

$n_artp(b, 7);$ (çıkışı: $71+7=78$ olur
3 değil!

İşaretçi \Rightarrow kodunu
inceleğin.