Kendisini çağıran bir fonksiyon özyinelemeli fonksiyon olarak bilinir. Ve bu teknik özyineleme olarak bilinir.

**Özyineleme nasıl çalışır?**

void recurse()

{

... .. ...

recurse();

... .. ...

}

int main()

{

... .. ...

recurse();

... .. ...

}

**Örnek: Özyinelemeyi Kullanan Doğal Sayıların Toplamı**

#include <stdio.h>

int sum(int n);

int main()

{

int number, result;

printf("pozitif bir tamsayı girin: ");

scanf("%d", &number);

result = sum(number);

printf("Toplam = %d", result);

return 0;

}

int sum(int num)

{

if (num!=0)

return num + sum(num-1); // sum() function calls itself

else

return num;

}

Çıktısı :

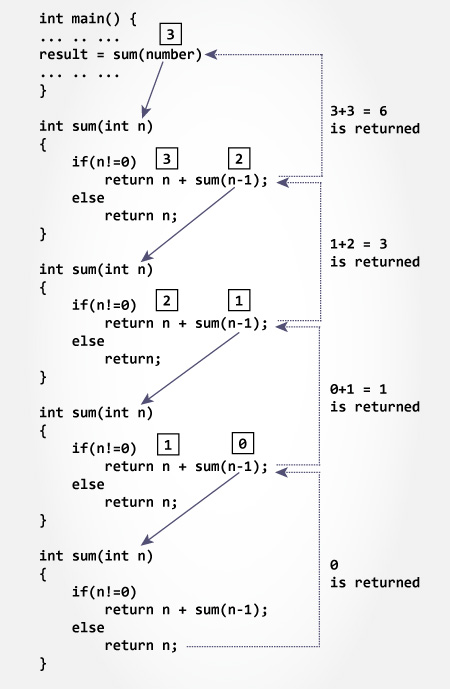
pozitif bir tamsayı girin:3

Toplam = 6

Başlangıçta, sum() fonksiyonu argüman olarak iletilen sayı ile birlikte main() fonksiyonundan çağrılır.

Farz edelim ki num değeri başlangıçta 3’tür. Bir sonraki işlev çağrısı sırasında 2 , sum() fonksiyonuna iletilir. Bu işlem, num 0’a eşit olana kadar devam eder.

Num 0’a eşit olduğunda, if koşulu başarısız olur ve diğer kısım tamsayıların toplamını main () işlevine döndürerek yürütülür.



**Özyinelemenin Avantajları ve Dezavantajları**  
Özyineleme, programı zarif ve daha okunaklı hale getirir. Ancak, projenizde performans daha önemli ise, özyineleme genellikle daha yavaş olduğu için döngüler kullanın.

Her özyinelemenin bir döngü olarak modellenebileceğini unutmayın.