C Programlama

Öğretim Elemanı Bilgileri

- Dr. Öğr. Üyesi Sema ATASEVER
- Nevşehir Hacı Bektaş Veli Üniversitesi, Mühendislik Mim.Fak. Bilgisayar Mühendisliği
- Web sayfası : https://biz.nevsehir.edu.tr/sema/tr
- Email : sema@nevsehir.edu.tr | s.atasever@gmail.com

Ölçme Yöntemi

ARA SINAV

- Ara sınav : 100 puan üzerinden değerlendirilecektir , Katkı : %40

FİNAL ÖDEVİ

- Final Sınavı : 100 puan üzerinden değerlendirilecektir , Katkı : %60
- Nihai ders notu hesabı : Ara sınavın %40'ı, Final notunun %60'ı alınarak hesaplanmaktadır!

9.Hafta Konuları

- Özyinelemeli Fonksiyonlar (Recursive Functions), örnek kod uygulamaları.

Özyineleme (recursive)

- C dili, özyinelemeli işlev olarak bilinen özelliği destekler.
- Özyinelemeli fonksiyonlar, problemleri özlü ve verimli bir şekilde çözmek için etkin bir şekilde kullanılabilirler.
- Genellikle, bir problemin çözümünün, aynı çözümü problemin alt kümelerine art arda uygulanmasıyla ifade edilebildiği uygulamalarda kullanılırlar.
- Diğer yaygın uygulamalar, ağaçlar ve listeler adı verilen veri yapılarının aranması ve sıralanmasını içerir. Özyinelemeli fonksiyonlar, en yaygın olarak, bir sayının faktöriyelini hesaplayan bir örnekle gösterilir. Bir pozitif tamsayının n faktöriyeli, n! , sadece 1'den n'ye kadar olan ardışık tamsayıların çarpımıdır. 0'ın faktöriyeli özel bir durumdur ve 1'e eşit olarak tanımlanır. Yani 5! şu şekilde hesaplanır:

```
5!      =   5 x 4 x 3 x 2 x 1
        =   120
```

Özyinelemeli Fonksiyonlar (Recursive Functions)

Faktöriyel Hesaplama Örneği:

```
int fact(int n)
{
    if (n < = 1) // base case
        return 1;
    else
        return n*fact(n-1);
}
```

```
= 5 * f(4)
= 5 * (4 * f(3))
= 5 * (4 * (3 * f(2)))
= 5 * (4 * (3 * (2 * f(1))))
= 5 * (4 * (3 * (2 * 1)))
= 5 * (4 * (3 * 2))
= 5 * (4 * 6)
= 5 * 24)
=120
```

https://www.geeksforgeeks.org/recursion/

Factorial of a Number Using Recursion

```c
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

Output

```
Enter a positive integer: 6
Factorial of 6 = 720
```

Özyinelemeli Faktöriyel Hesaplama Örneği:

```c
#include <stdio.h>
int main (void)
{
    unsigned int j;
    unsigned long int factorial (unsigned int n);

    for ( j = 0; j < 11; ++j )
        printf ("%2u! = %lu\n", j, factorial (j));
    return 0;
}
// Recursive function to calculate the factorial of a positive integer
unsigned long int factorial (unsigned int n)
{
    unsigned long int result;
    if ( n == 0 )
        result = 1;
    else
        result = n * factorial (n - 1);

    return result;
}
```

Faktöriyel Hesaplama

- Pozitif bir n sayısının faktöriyeli şu şekilde verilir:
  - factorial of n (n!) = 1 * 2 * 3 * 4....n
- Negatif bir sayının faktöriyeli yoktur. Ve 0'ın faktöriyeli 1'dir.

Factorial of a Number

```c
#include <stdio.h>
int main() {
    int n, i;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);

    // shows error if the user enters a negative integer
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else {
        for (i = 1; i <= n; ++i) {
            fact *= i;
        }
        printf("Factorial of %d = %llu", n, fact);
    }

    return 0;
}
```

Output

```
Enter an integer: 10
Factorial of 10 = 3628800
```

https://www.programiz.com/c-programming/examples/factorial

Özyineleme kullanarak bir Sayının Asal olup olmadığını bulan C kod örneği

```c
#include<stdio.h>

// declaring the recursive function
int isPrime(int, int);

int main()
{
    int num, prime;
    printf("Enter a positive number to check if Prime: ");
    scanf("%d", &num);
    prime = isPrime(num, num/2);
    if(prime == 1)
    {
        printf("\n\n%d is a prime number\n\n", num);
    }
    else
    {
        printf("\n\n%d is a Composite number\n\n", num);
    }
    return 0;
}

// function definition
int isPrime(int n, int i)
{
    if(i == 1)
        return 1;    // return statement terminates the recursive function
    else
    {
        if(n%i == 0)
            return 0;
        else
            isPrime(n, i-1);     // recursive call not using return statement
    }
}
```

https://www.studytonight.com/c/programs/recursion/prime-or-composite-using-recursion#

Özyineleme kullanarak birden girilen sayıya kadar olan doğal sayıların toplamını bulan C kod örneği

Example: Sum of Natural Numbers Using Recursion

```c
#include <stdio.h>
int sum(int n);

int main() {
    int number, result;

    printf("Enter a positive integer: ");
    scanf("%d", &number);

    result = sum(number);

    printf("sum = %d", result);
    return 0;
}

int sum(int n) {
    if (n != 0)
        // sum() function calls itself
        return n + sum(n-1);
    else
        return n;
}
```

Output

```
Enter a positive integer:3
sum = 6
```

https://www.programiz.com/c-programming/c-recursion