

ALGORİTMA VE PROGRAMLAMA I

Yrd. Doç. Dr. Deniz KILINÇ
deniz.kilinc@cbu.edu.tr

Genel Bakış...

2

- Giriş
- Esnek Argümanlı Fonksiyonlar
- `main()` Fonksiyonuna Parametre Aktarımı

9. BÖLÜM

3

Esnek Argümanlı Fonksiyonlar ve main() Fonksiyonuna Parametre Aktarımı

Giriş

4

- Aşağıdaki gibi üçüncü dereceden bir polinomu ele alalım:

$$P(x) = a + bx + cx^2 + dx^3$$

- a, b, c, d katsayıları gerçel sayı sabitleridir.
- x ise gerçel sayı türündeki bir değişkendir.

Bu polinomu temsil eden basit bir fonksiyonu nasıl tanımlarız?

Esnek Argümanlı Fonksiyonlar

5

$$P(x) = a + bx + cx^2 + dx^3$$

```
float p(float x, float a, float b, float c, float d)
{
    float p = a + b*x + c*x*x + d*x*x*x;
    return p;
}
```

Esnek Argümanlı Fonksiyonlar (devam....)

6

- Buna göre, $x = 1.7$ 'de,

$$P(x) = 1 - 2x$$

değerini hesaplamak için bu fonksiyon aşağıdaki gibi çağırılmalıdır:

$$sonuc = p(1.7, 1.0, -2.0, 0.0, 0.0);$$

- Burada, kullanılmayan katsayılar için **0.0** değeri mutlaka fonksiyona geçirilmelidir.

Esnek Argümanlı Fonksiyonlar (devam....)

7

- Kullanılmayan argümanlar geçirilmeden de fonksiyonu çağırarak mümkündür.
- C Programlama dili, kullanıcılarına argümanları esnek olarak geçirme imkanı verir.
- Bunun anlamı, belli kurallar sağlandığında, **p()** fonksiyonunun aşağıdaki gibi çağrılabilmesidir:

```
/* x a b */  
sonuc = p(1.7, 1.0, -2.0);
```

Esnek Argümanlı Fonksiyonlar (devam....)

8

- Esnek argümanlar için 2 temel kural vardır:
 1. Esnek argümanlar kullanımı isteğe bağlıdır.
 2. Esnek argümanları oluşturan küme ardışık olarak listeye eklenmelidir.
- Bu türden argümanlar, aşağıdaki gibi, fonksiyonun parametre listesi kısmında ... (3 nokta) ile belirtilir.

```
double p(double x, int n, ...)  
{  
  
}
```


Esnek Argümanlı Fonksiyonlar (devam....)

9

- Esnek argümanlı fonksiyon tanımlaması yapabilmek için **stdarg.h** kütüphanesinde üç tane makro fonksiyon tanımlanmıştır.

| Tip / Fonksiyon | Açıklama |
|-----------------|--|
| va_list | ardışık esnek argümler için tip belirleyici |
| va_start(ap, n) | va_list tipinde bildirilmiş ap göstericisi için <u>bellekten n elemanlı yer ayırır.</u> |
| va_arg(ap, tip) | Veri tipi tip ile belirlenmiş küme elemanlarına erişimi sağlar. |
| va_end(ap) | va_list tipinde bildirilmiş ap göstericisi için <u>bellekten bölgeyi boşaltır.</u> |

Örnek1: n Tane Sayının Toplamı

10

- Öyle bir fonksiyon yazalım ki istediğimiz kadar sayıyı fonksiyona girdi (argüman) olarak verebilelim ve sonucunda bu sayıları toplasın.
- Kaç sayı olacak ? 3 sayı? 5 sayı ?
- Normalde bu bilgi önemli mi? Cevap: Evet
- Ancak Esnek argümanlı fonksiyon ile argüman sayısı önemsiz hale gelecek.

Örnek1: n Tane Sayının Toplamı (devam...)

11

```
#include <stdarg.h>
#include <stdio.h>

int toplar(int, ...);

int main(void)
{
    printf("toplara(2, 11,22)           = %d\n", toplar(2, 11,22));
    printf("toplara(3, 11,22,33)        = %d\n", toplar(3, 11,22,33));
    printf("toplara(4, 11,22,33,44)     = %d\n", toplar(4, 11,22,33,44));
    printf("toplara(5, 11,22,33,44,55)  = %d\n", toplar(5, 11,22,33,44,55));
    printf("toplara(6, 11,22,33,44,55,66) = %d\n", toplar(6, 11,22,33,44,55,66));

    return 0;
}

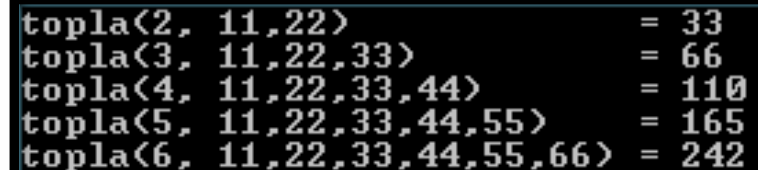
int toplar(int n, ...)
{
    va_list ap;
    int i, top = 0;

    va_start(ap, n);

    for (i=1; i<=n; i++)
        top += va_arg(ap, int);

    va_end(ap);
    return top;
}
```

I



```
toplara(2, 11,22)           = 33
toplara(3, 11,22,33)        = 66
toplara(4, 11,22,33,44)     = 110
toplara(5, 11,22,33,44,55)  = 165
toplara(6, 11,22,33,44,55,66) = 242
```

Örnek2: n Tane Sayının En Küçüğünü Bulan Fonksiyon

12

- Öyle bir fonksiyon yazalım ki istediğimiz kadar sayıyı fonksiyona girdi (argüman) olarak verebilelim ve geriye değer olarak bu sayılardan en küçüğünü dönsün.

Örnek2: n Tane Sayının En Küçüğünü Bulan Fonksiyon (devam...)

13

```
#include <stdarg.h>
#include <stdio.h>

int minimum_bul(int arg_sayac, ...)
{
    int i;
    int minimum_sayi, a;

    va_list ap;
    va_start(ap, arg_sayac);
    minimum_sayi = va_arg(ap, int);

    for(i = 2; i <= arg_sayac; i++) {
        if((a = va_arg(ap, int)) < minimum_sayi)
            minimum_sayi = a;
    }

    va_end(ap);

    return minimum_sayi;
}

int main()
{
    int sayi_adi = 5;
    printf("En küçük sayı: %d", minimum_bul(sayi_adi, 12, 2, 6, 7, 100));
    getchar();
    return 0;
}
```

main() Fonksiyonuna Parametre Aktarımı

14

- Ana programa yani main() fonksiyonuna parametre aktarımı, derlenmiş (çalıştırılabilir) bir program komut satırından (işletim sistemi ortamından) çalıştırılacağı zaman yapılır.
- Parametre aktarımı, programın adı yazılıp bir boşluk bırakıldıktan hemen sonra yapılır.
- Parametreler, komut satırından sayısal olarak girilse bile program içinde karakter topluluğu (string) olarak gelir.
- Bu durumda, bu ifadeleri sayısal değerlere çeviren (**atoi()**, **atol()**, **atof()** gibi) fonksiyonlar kullanılır.

Örnek1: Komut Satırından Girilen 2 Sayının Toplamı

15

```
#include <stdio.h>
#include <stdlib.h>

int main(int argsay, char *argvek[])
{
    int toplam;
    if(argsay < 3){
        printf("Eksik parametre !\n");
        exit(1);
    }

    if(argsay > 3){
        printf("Cok fazla parametre !\n");
        exit(1);
    }

    toplam = atoi(argvek[1]) + atoi(argvek[2]);

    printf("Toplamlari %d\n", toplam);

    return 0;
}
```

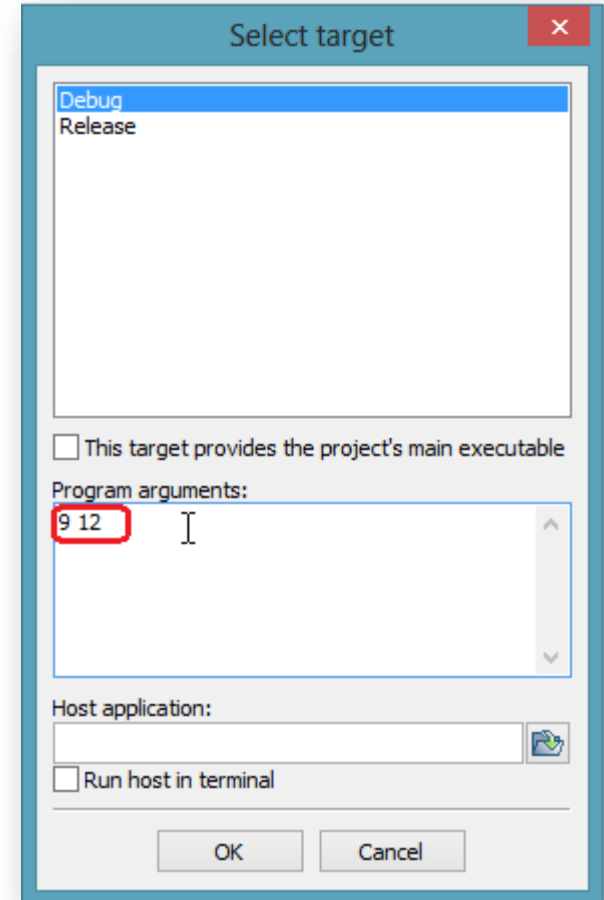
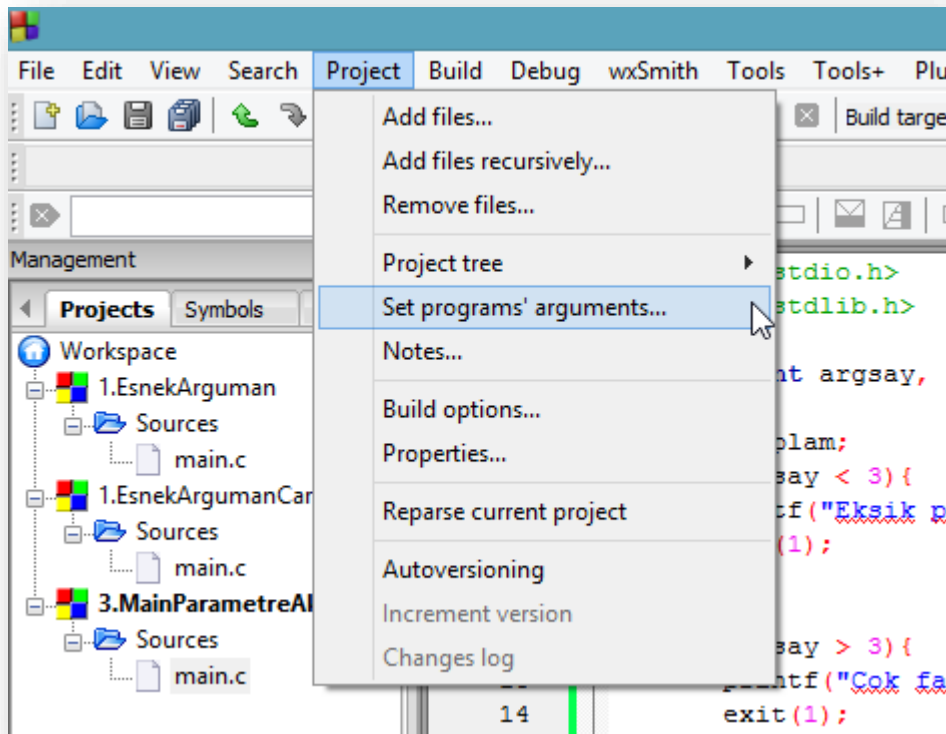
Run veya Ctrl+F10



```
Eksik parametre !
Process returned 1 (0x1) e>
Press any key to continue.
-
```

Örnek1: Komut Satırından Girilen 2 Sayının Toplamı (devam...)

16



Örnek1: Komut Satırından Girilen 2 Sayının Toplamı (devam...)

17

```
#include <stdio.h>
#include <stdlib.h>

int main(int argsay, char *argvek[])
{
    int toplam;
    if(argsay < 3){
        printf("Eksik parametre !\n");
        exit(1);
    }

    if(argsay > 3){
        printf("Cok fazla parametre !\n");
        exit(1);
    }

    toplam = atoi(argvek[1]) + atoi(argvek[2]);

    printf("Toplamlari %d\n", toplam);

    return 0;
}
```

Run veya Ctrl+F10

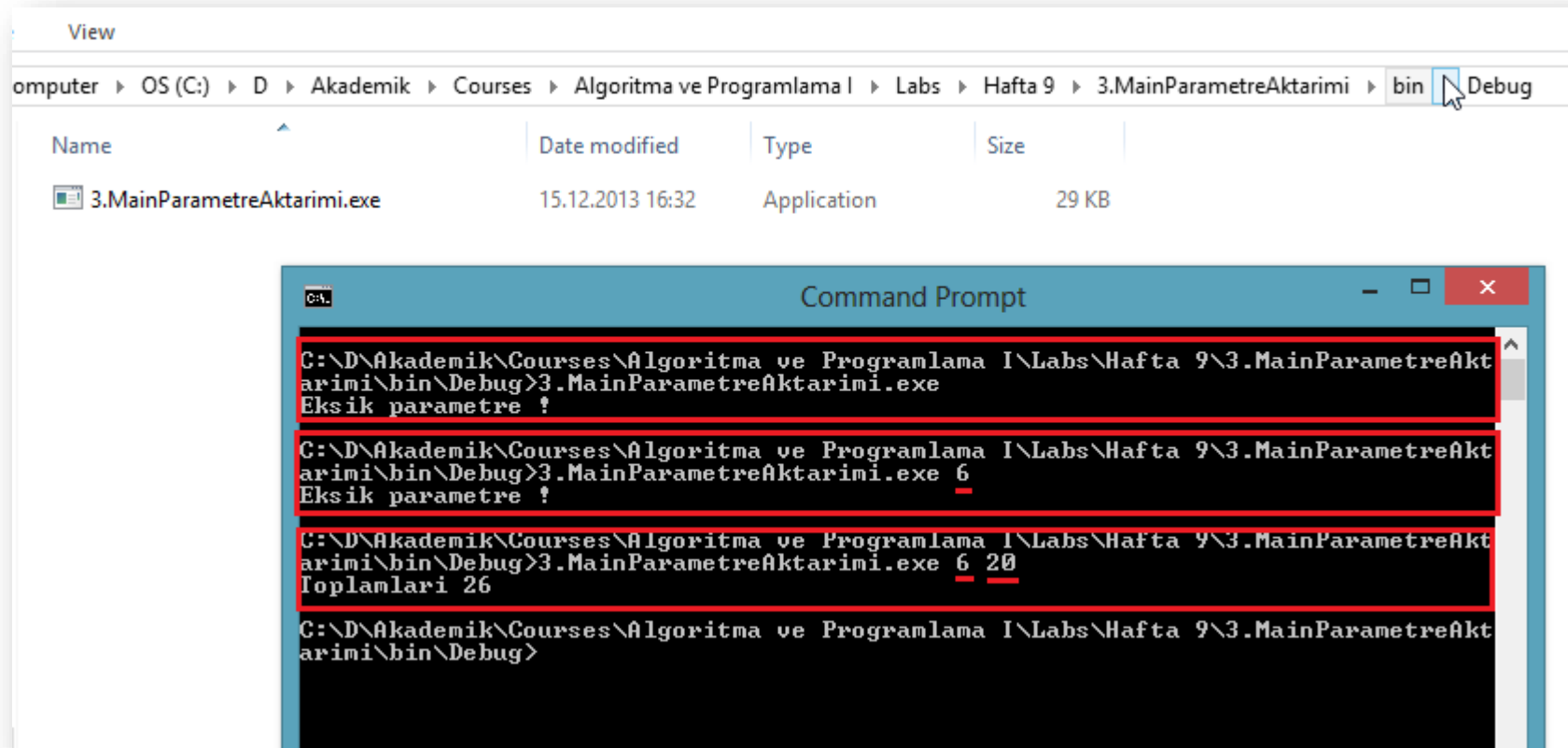


```
Toplamlari 21
Process returned 0 (0x0)   executio
Press any key to continue.
```

Örnek1: Komut Satırından Girilen 2 Sayının Toplamı (devam...)

18

- Ya da Windows işletim sistemi için DOS komutu satırına geçilir ve ilgili exe uygulaması bulunarak aşağıdaki gibi program çalıştırılabilir:



The screenshot shows a Windows File Explorer window with the path: Computer > OS (C:) > D > Akademik > Courses > Algoritma ve Programlama I > Labs > Hafta 9 > 3.MainParametreAktarimi > bin > Debug. The file 3.MainParametreAktarimi.exe is listed with a date modified of 15.12.2013 16:32, type Application, and size 29 KB.

Below the File Explorer window is a Command Prompt window titled "Command Prompt" showing the execution of the program 3.MainParametreAktarimi.exe with various command-line arguments. The output shows the sum of the two numbers entered as arguments.

```
C:\D\Akademik\Courses\Algoritma ve Programlama I\Labs\Hafta 9\3.MainParametreAktarimi\bin\Debug>3.MainParametreAktarimi.exe
Eksik parametre !

C:\D\Akademik\Courses\Algoritma ve Programlama I\Labs\Hafta 9\3.MainParametreAktarimi\bin\Debug>3.MainParametreAktarimi.exe 6
Eksik parametre !

C:\D\Akademik\Courses\Algoritma ve Programlama I\Labs\Hafta 9\3.MainParametreAktarimi\bin\Debug>3.MainParametreAktarimi.exe 6 20
Toplamlari 26

C:\D\Akademik\Courses\Algoritma ve Programlama I\Labs\Hafta 9\3.MainParametreAktarimi\bin\Debug>
```

KAYNAKLAR

19

- N. Ercil Çağıltay ve ark., C DERSİ PROGRAMLAMAYA GİRİŞ, Ada Matbaacılık, ANKARA; 2009.
- Milli Eğitim Bakanlığı "Programlamaya Giriş ve Algoritmalar Ders Notları", 2007
- <http://tr.wikipedia.org/wiki/Code::Blocks>
- <http://www.codeblocks.org>
- <http://www.AlgoritmaveProgramlama.com>
- <http://www1.gantep.edu.tr/~bingul/c>



Algoritma ve Programlama

İYİ ÇALIŞMALAR...

Yrd. Doç. Dr. Deniz KILINÇ
deniz.kilinc@cbu.edu.tr