BSM101 Programlama Dilleri I

Hafta 12
Karakter Tutan Diziler

Doç. Dr. Caner ÖZCAN

Katar (String) Tanımlama

- Dizileri ve çok boyutlu dizileri gördük.
- Katar dediğimiz şey de aslında bir dizidir.
- Değişken tipi char yani karakter olan diziler, 'katar' ya da İngilizce adıyla 'string' olarak isimlendirilirler.
- Örneğin bir tam sayı (int) dizisinde, tam sayıları saklarken; bir karakter dizisinde -yani katarda- karakterleri (char) saklarız.
- İsimler, adresler, kullanıcı adları, telefonlar vs... sözle ifade edilebilecek her şey için karakter dizilerini kullanırız.

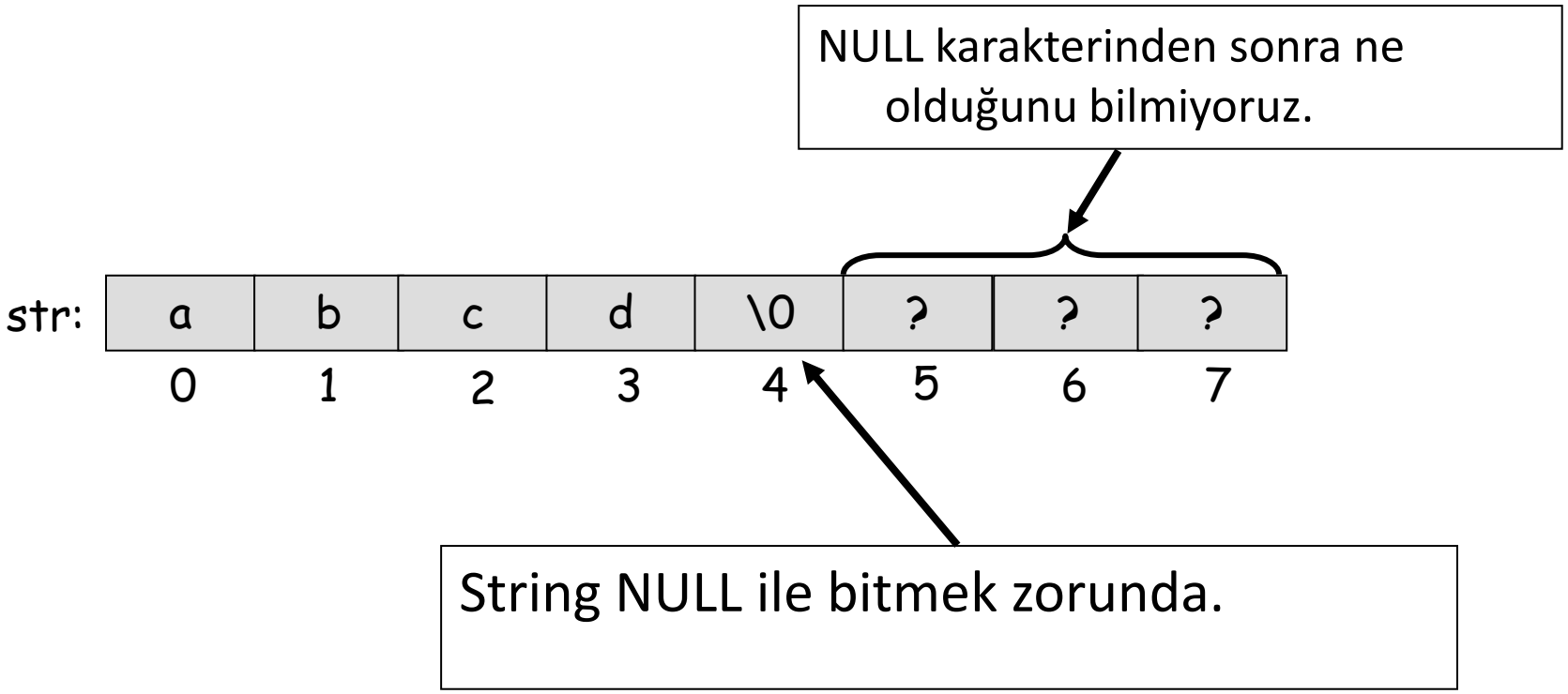
String Tanımlama

- String NULL karakter ‘\0’ ile biten bir karakter dizisidir.

- Örnek: char str[8];
  - En çok 8 karakter alabilen bir dizi oluşturur.
  - Eğer str dizisi string olarak kullanılacak ise en fazla 7 karakter alabilir ve sonu NULL karakter ‘\0’ ile bitmek zorunda.

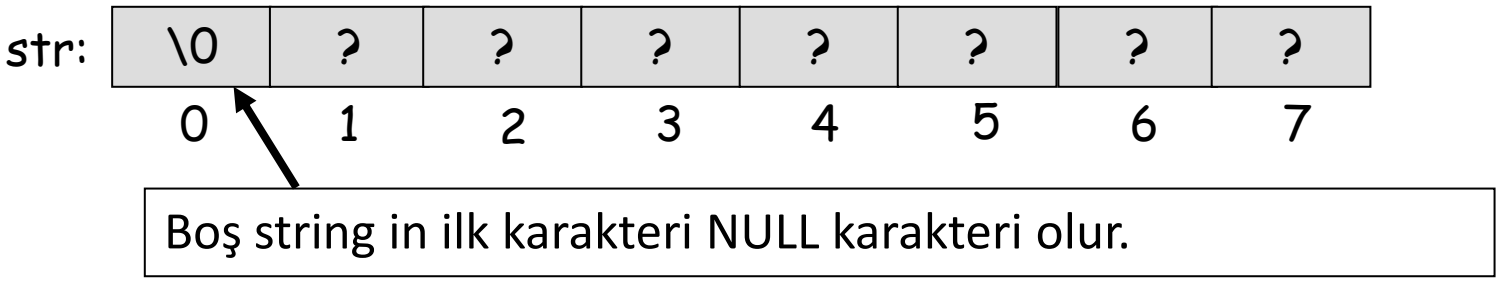
String Gösterimi

- Eğer str de “abcd” yazısını depolarsak bu aşağıdaki şekilde görünecektir.



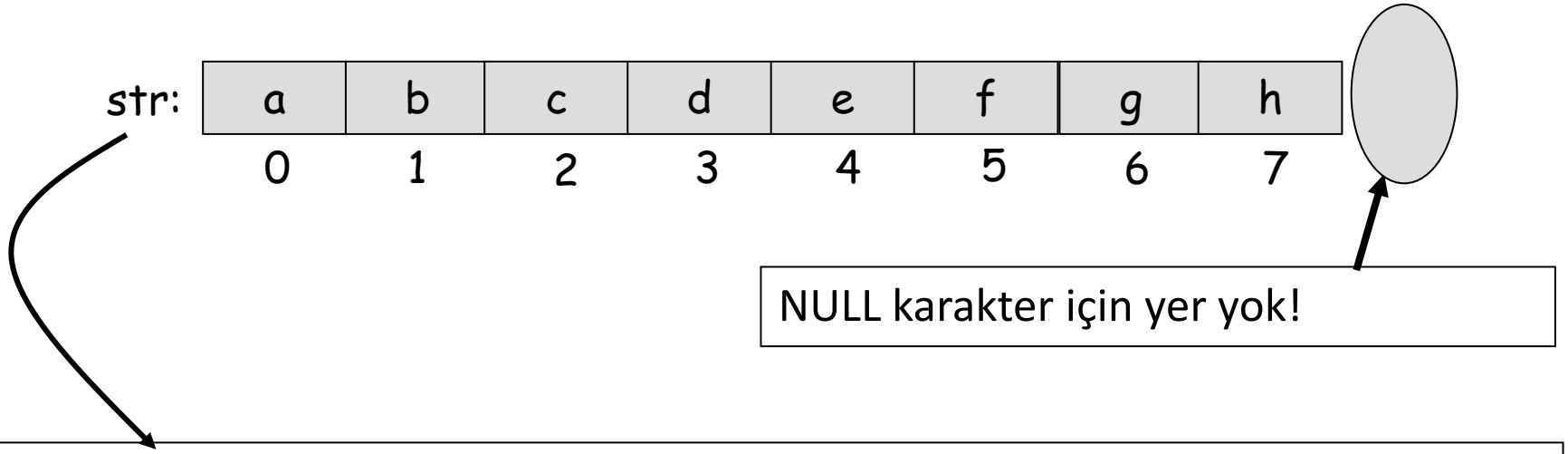
Boş String

- Boş string “” ilk elemanı NULL karakter ‘\0’ olan bir karakter dizisini ifade eder.



String Maksimum Uzunluğu

- 8 karakter uzunluğunda bir string örneğin, “abcdefgh” str de depolanamaz



- Bu 8 karakter içeren bir karakter dizisidir.
- Fakat string DEĞİL. Bir string her zaman NULL karakter ile bitmek ZORUNDA!

String: UYARI

- Sadece tekrar, char str[8] gibi bir beyan basitçe str de en fazla 8 karakter saklayabileceğimizi vurgular.

- Programın çalışması sırasında herhangi bir noktada str de 8 den daha fazla karakter saklamak isteyebiliriz.

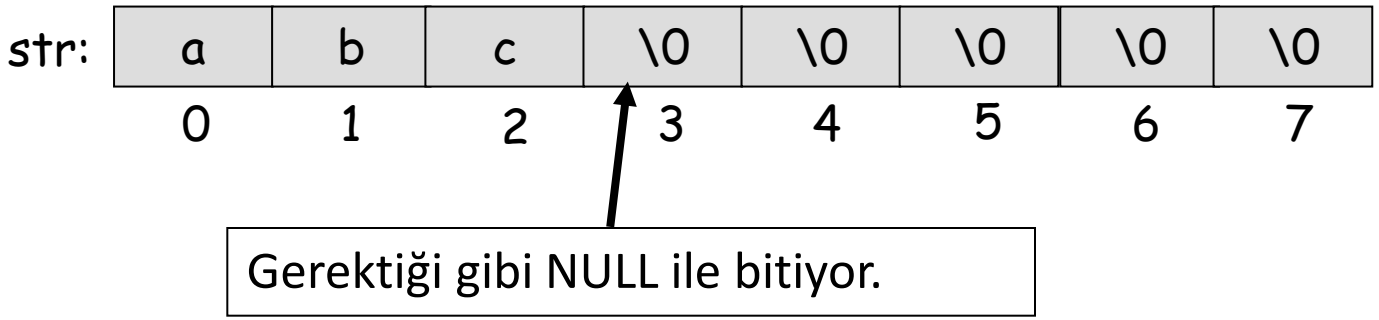
- Fakat eğer “str” bir string depoluyorsa en fazla 8-1=7 karakter saklayabiliriz, ve her zaman NULL ile bitmek zorunda.

String: İlk Değer Atama

- Bir karakter dizisi tanımlama anında diğer dizilerde olduğu gibi aşağıdaki gibi tanımlanabilir.

```
char str[8] = {'a', 'b', 'c'};
```

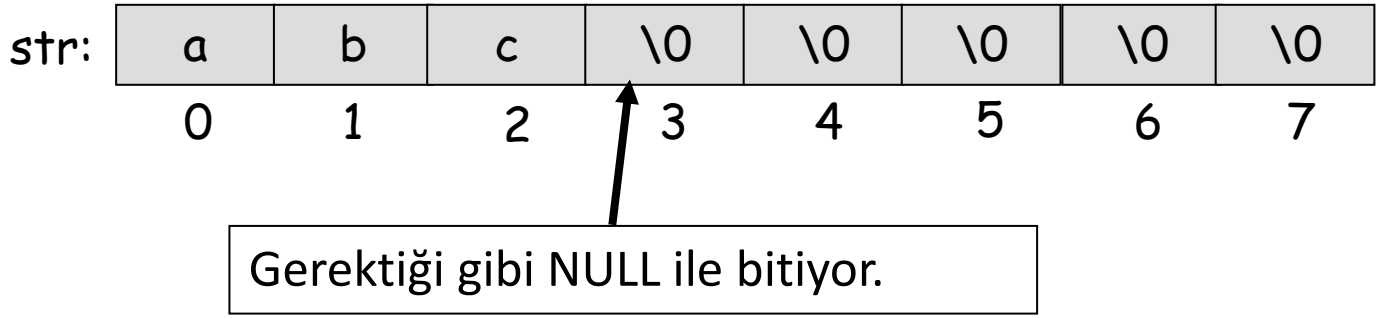
- Hatırlayın belirtilmeyen elemanlar ' \0 ' ile dolduruluyordu, ki bu da NULL karakter oluyor.
  - Bu nedenle yukarıdaki beyan aşağıdaki string e karşılık geliyor.



String: İlk Değer Atama

- Eğer bir karakter dizisi bir string depolayacaksa, aşağıdaki gibi basitçe ilk değer atanabilir.
  - Sadece string çift tırnak içine konulur. Buna string literali denir.

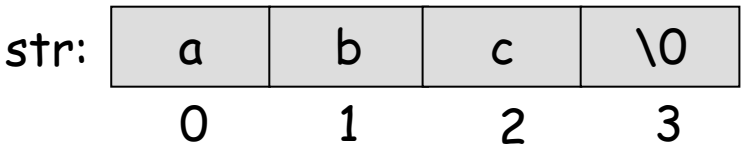
```
char str[8] = "abc"; /* önceki ile aynı*/
```



String: İlk Değer Atama

- Eğer dizinin uzunluğunu tanımlama anında belirtmiyorsak derleyici string uzunluğu + NULL karakter kadar yer ayırır.

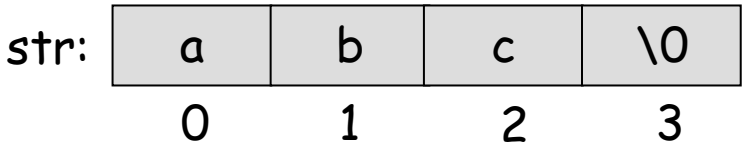
```
char str[] = "abc";
```



String: İlk Değer Atama

- String ler genellikle aşağıdaki gibi tanımlanır.

```
char *str = "abc";
```



- Bu tanımlama ile önceki tanımlamalar arasındaki fark: bu yol ile tanımlanan string ler READ-ONLY oluyor ve değiştirilemiyorlar.
- char str[]="abc"; şeklinde tanımlanan string leri istediğiniz gibi değiştirebilirsiniz.

String Yazdırmak

- C string leri yazdırmak için iki fonksiyon sunuyor.
  - (1) puts(str); (2) printf("%s", str);

```
char str1[]="bu benim ilk stringim";

/* stringi yazdırır ve imleç sonraki satırın başına geçer.*/
puts(str1);

/* stringi imlecin olduğu yerden yazdırmaya başlar */
printf("%s", str1);

/* 40 boşluk ayırır ve stringi bunun içinde sağa dayalı
   olarak yazdırır. */
printf("%40s", str1);

/* 40 boşluk ayırır ve stringi bunun içinde sola dayalı
   olarak yazdırır. */
printf("%-40s", str1);
```

String Yazdırmak

```
char str1[]="bu benim ilk stringim";

/* stringden sadece ilk 10 karakteri yazar,
 * sağa dayalı */
printf("%.10s", str1);

/* 40 boşluk ayırır ve sadece ilk 10 karakteri yazdırır,
   * sağa dayalı */
printf("%40.10s", str1);

/* 40 boşluk ayırır ve sadece ilk 10 karakteri yazdırır,
   * sola dayalı */
printf("%-40.10s", str1);
```

String Okumak

- Klavyeden string almak için C iki fonksiyon sunuyor.
  - (1) gets(str); (2) scanf("%s", str);

```
char str2[80];

/* '\n' girilene kadar girilen stringi okur. */
gets(str2);

/* bütün boşluk karakterlerini (space, tab, newline) geçerek
    * girileni sonraki boşluk karakterine kadar okur.*/
scanf("%s", str2);
```

String Okumak

```
char str2[80];

/* bütün boşluk karakterlerini (space, tab, newline) geçerek
    * girileni sonraki boşluk karakterine kadar okur.*/
scanf("%s", str2);

/*eğer giriş aşağıdaki gibi ise:_ space olarak farz edelim */
_ _xyz123_ _ _45_ _67
```

- scanf ilk iki boşluğu geçecek ve str2 “xyz123” olacak.
- Sonra boşluğu görecek ve okuma duracak.
- Bir sonraki scanf(“%s”, …) bu boşlukları geçecek ve “45” i okuyacak.

String Okumak

- İsterseniz "Enter" girilene kadar girişi okuyacak kendi okuma fonksiyonunuzu yazabilirsiniz.

```
char *ReadLine(char *str){
  char ch;
  char *p = str;
  while((ch=getchar()) != '\n')
     *p++=ch;
  *p = '\0';  /* The end of the string is ended with NULL
   characters. */
  return str;
} /* end-ReadLine */

main(){
  char str[80];

  ReadLine(str);
  printf("Entered row= <%s>\n", str);
} /* end-main */
```

String Okumak

- Bir başka versiyon "Enter" girilene kadar VEYA “n” adet karakter girilene kadar olabilir.

```
char *ReadNLine(char *str, int n){
  char ch;
  char *p = str;
  while (n-- > 0){
    if ((ch = getchar()) == '\n') break;
    *p++ = ch;
  } /* end-while */
  *p = '\0'; /* stringin sonunu NULL karakter yap */
  return str;
} /* end-ReadNLine */

main(){
  char str[80]; char *p = NULL;
    p = ReadNLine(str, 79); /* can get a maximum of 79
   characters */
  printf("Entered row= <%s>\n", p);
} /* end-main */
```

String İşlemleri

- C standard kütüphanesi string leri manipüle etmek için birçok fonksiyon içeriyor.
  - Bu fonksiyonları kullanmak için <string.h> dosyasını eklemeniz gerekiyor. #include <string.h>
- Bazı önemli fonksiyonlar:
  - strlen(const char *str);
  - strcpy(char *str1, const char *str2);
  - strcat(char *str1, const char *str2);
  - strcmp(const char *str1, const char *str2);
- Bu fonksiyonların detaylarına önümüzdeki dönem gireceğiz.

Örnek: Karakter Dizisinin Uzunluğu

```
#include <stdio.h>
int main(void){
   char s[40];
   int  k = 0;

   /* diziyi oku */
   printf("Bir seyler yazin : ");
   gets(s);

   /* sonlandırıcı karaktere kadar karakterleri say */
   while( s[k]!='\0' )
       k++;
    printf("Dizinin uzunlugu : %d\n",k);

 return 0;
}
```

Örnek: Karakter Dizisinin Tersi

```
#include <stdio.h>
int main(void){
   char s[40], gecici;
   int  i, n;
   /* diziyi oku */
   printf("Bir seyler yazin : ");
   gets(s);
   /* sonlandırıcı karaktere kadar */
   for(n=0; s[n] != '\0'; n++);
   for(i=0; i<n/2; i++){
      gecici   = s[n-i-1];
      s[n-i-1] = s[i];
      s[i]     = gecici;
   }
   printf("Tersi           : %s\n",s);
 return 0;
}
```

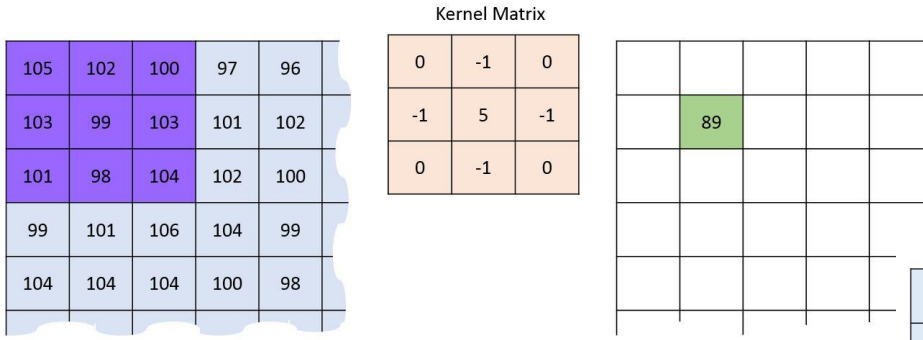
HOMEWORK
NEXT

ÖDEV (HOMEWORK)

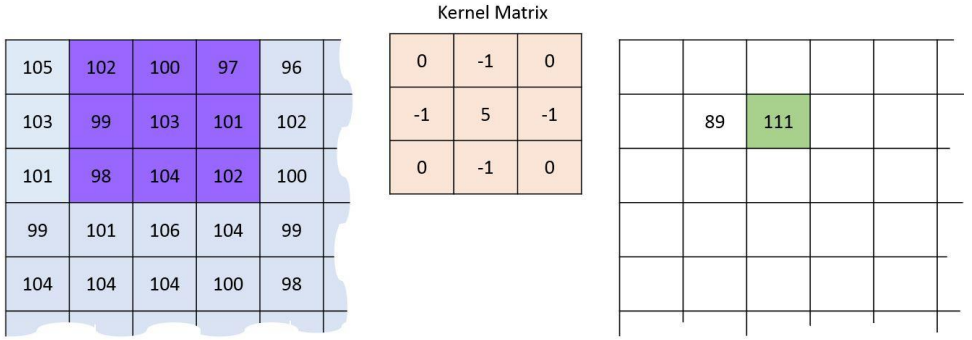
Soru 1: Ödevin 15 puanlık bölümünü oluşturmaktadır. (**Question 1:** It consists of 15 points of the homework.)

Aşağıdaki gibi f isimli iki boyutlu MxN boyutlu bir matrisi $0-255$ arasında rasgele değerler alacak şekilde oluşturunuz. Hazırlanan bu f matrisi, $3x3$ boyutlu ve değerleri dışarıdan girilecek bir v filtresi ile iki boyutlu filtre işlemine tabi tutulacaktır. Elde edilen sonuçlar iki boyutlu g matrisinde saklanacaktır. Filtre işlemi f matrisinin kenar elemanları hariç tüm elemanları üzerine uygulanacaktır. Örnek bir hesaplama aşağıda verilmektedir. Bu işlemleri gerçekleştiren C kodunu yazınız.

(Create a two-dimensional MxN-sized matrix named f, which takes random values from $0-255$. This matrix f will be subjected to a two-dimensional filter process with a filter v whose $3x3$ dimensional values will be entered from keyboard. The results will be stored in a two-dimensional g matrix. The filter process will be applied to all elements of the matrix f except the edge elements. An example calculation is given below. Write the C code that performs these operations.)



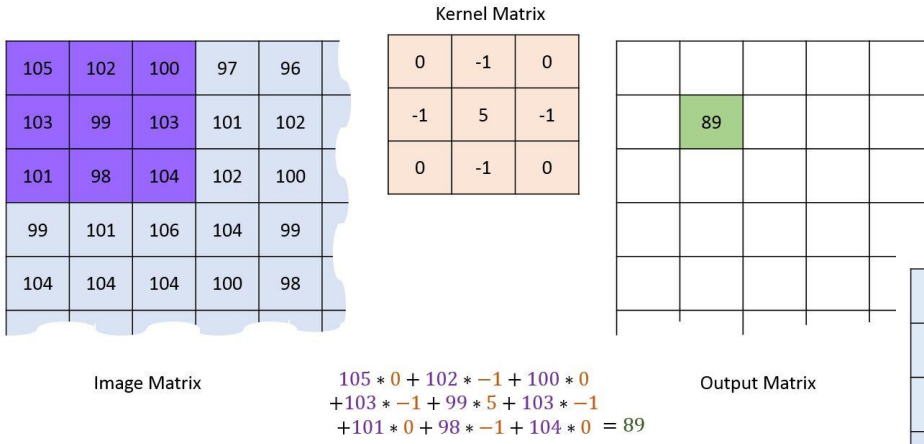
$$105*0+102*-1+100*0+103*-1+99*5+103*-1+101*0+98*-1+104*0=89$$



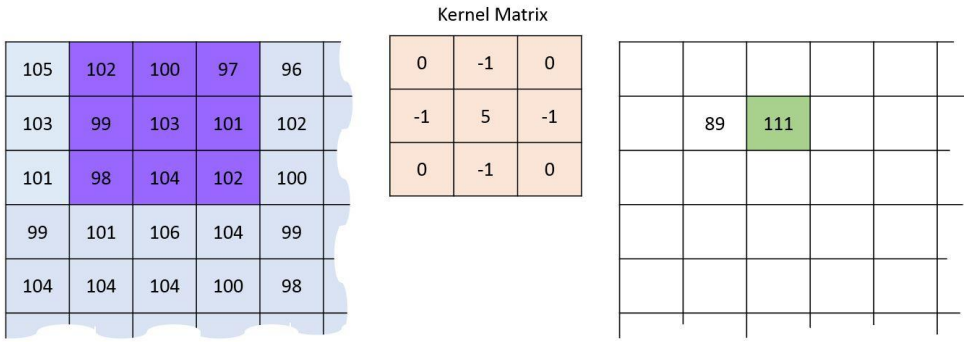
$$102*0+100*-1+97*0+99*-1+103*5+101*-1+98*0+104*-1+102*0=111$$

Soru 2: Ödevin 10 puanlık bölümünü oluşturmaktadır. (**Question 2:** It consists of 10 points of the homework.)

Birinci soruda hazırlanan algoritma kenar elemanlarına uygulanmamaktaydı. Bu adımda v filtresini f matrisinin kenar elemanlarına da uygulayabilecek bir çözüm geliştiriniz. Çözümlerinizde tamamen özgünlük aranacak olup değerlendirmeler buna göre yapılacaktır.
(The algorithm prepared in the first question was not applied to the edge elements. In this step, develop a solution that can also apply the filter v to the edge elements of the matrix f. Completely originality will be expected in your solutions and evaluations will be made accordingly.)

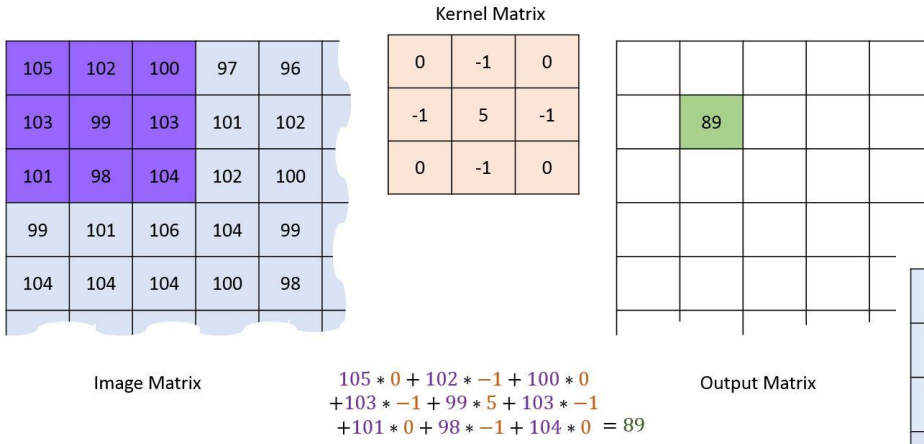


$$105*0+102*-1+100*0+103*-1+99*5+103*-1+101*0+98*-1+104*0=89$$

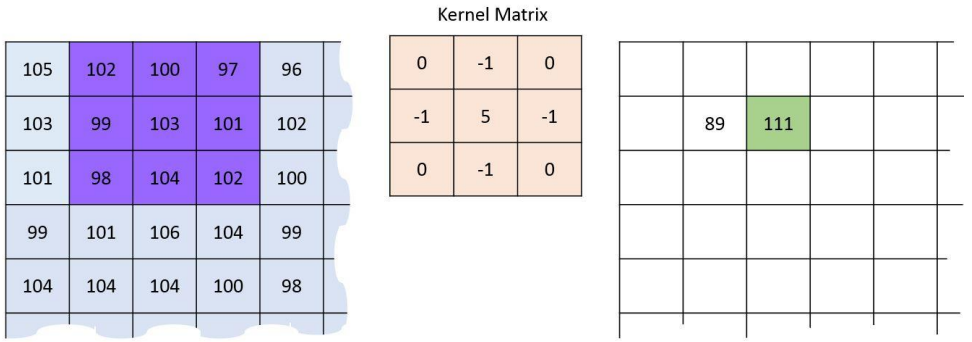


$$102*0+100*-1+97*0+99*-1+103*5+101*-1+98*0+104*-1+102*0=111$$

Not: Ödevin gösterim tarihi 16-20 Aralık tarihlerinde laboratuvar dersleridir. Ödev e-mail veya başka bir şekilde gönderilmeyecek, yüz yüze bilgisayar başında anlatılacaktır. Bütün öğrenciler kayıt oldukları laboratuvar saatinde ödevini gösterecektir. Bunun haricinde ödev kontrolü yapılmayacaktır. Ödev 25 puandır. Ödevden alınan not final ve bütünleme sınav notuna etki edecektir. **Kopya ödevler -15 puan olarak değerlendirilecektir.**
(**Note:** The deadline for the assignment is laboratory lectures on December 16-20. The assignment will not be sent by 3-mail or in any other way. The homework will be explained face to face at the computer. All students will show their homework during the laboratory hours. Homework control will not be done by another way or another time. The assignment is 25 points. The grade from the assignment will affect the final and make-up exam grade. **Copy assignments will be evaluated as -15 points.**)



$$105*0+102*-1+100*0+103*-1+99*5+103*-1+101*0+98*-1+104*0=89$$



$$102*0+100*-1+97*0+99*-1+103*5+101*-1+98*0+104*-1+102*0=111$$

Kaynaklar

- Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- Paul J. Deitel, “C How to Program”, Harvey Deitel.
- Bayram AKGÜL, C Programlama Ders notları