

BMB204. Veri Yapıları

Ders 2.

Veri Yapıları (Data Structures)

Dersin Planı

- Bilgisayarda veri kavramı
 - İkili, BCD, ASCII ayrımı
- Veri türleri
 - Char, String, Int, Double vs.
 - Diziler
- Encoding (Kodlama)
 - ASCII, UTF8

Bilgisayarda Veri Kavramı

- Veri ham olarak 1 ve 0'lardan oluşan bir bit (binary digit) dizisi şeklindedir.
- Ve bu bit dizileri çeşitli formatlarda (resim, sayı, word, excel, vs..) anlaşılır veri olarak olarak karşımıza çıkar.

16 Bitlik bir bit dizisi	0100001001000010
ASCII çevrimi (American Standart Code for Information Interchange)	0100001001000010 B A
BCD (Binary Coded Decimal) Çevrimi	0100001001000010 4 2 4 2
İşaretsiz Tamsayı ise (16 bitlik)	0100001001000010 16961

İşlemci ve diskteki veri boyutları

Processor or Virtual Storage

- 1 Bit = Binary Digit
- 8 Bits = 1 Byte
- 1024 Bytes = 1 Kilobyte
- 1024 Kilobytes = 1 Megabyte
- 1024 Megabytes = 1 Gigabyte
- 1024 Gigabytes = 1 Terabyte
- 1024 Terabytes = 1 Petabyte
- 1024 Petabytes = 1 Exabyte
- 1024 Exabytes = 1 Zettabyte
- 1024 Zettabytes = 1 Yottabyte
- 1024 Yottabytes = 1 Brontobyte
- 1024 Brontobytes = 1 Geopbyte

Disk Storage

- 1 Bit = Binary Digit
- 8 Bits = 1 Byte
- 1000 Bytes = 1 Kilobyte
- 1000 Kilobytes = 1 Megabyte
- 1000 Megabytes = 1 Gigabyte
- 1000 Gigabytes = 1 Terabyte
- 1000 Terabytes = 1 Petabyte
- 1000 Petabytes = 1 Exabyte
- 1000 Exabytes = 1 Zettabyte
- 1000 Zettabytes = 1 Yottabyte
- 1000 Yottabytes = 1 Brontobyte
- 1000 Brontobytes = 1 Geopbyte

bits (b) or bytes (B)

Decimal			Binary		
Value	Metric		Value	JEDEC	IEC
1000	k	kilo	1024	K	kilo Ki kibi
1000 ²	M	mega	1024 ²	M	mega Mi mebi
1000 ³	G	giga	1024 ³	G	giga Gi gibi
1000 ⁴	T	tera	1024 ⁴	-	- Ti tebi
1000 ⁵	P	peta	1024 ⁵	-	- Pi pebi
1000 ⁶	E	exa	1024 ⁶	-	- Ei exbi
1000 ⁷	Z	zetta	1024 ⁷	-	- Zi zebi
1000 ⁸	Y	yotta	1024 ⁸	-	- Yi yobi

Programlama Dillerinde Veri Yapılarının Sınıflandırılması

- Veri yapıları biri temel, diğeri tanımlamalı olmak üzere iki sınıf altında toplanabilirler.
 - Temel veri yapıları: doğrudan değişken veya sabit bildirimi yapılarak kullanılır.
 - `int x;`
 - `char kr;`
 - `float kesirli;`
 - Tanımlamalı veri yapıları: kendisinden önceki tanımlamalı veya temel veri yapıları üzerine kurulurlar. Başka bir deyişle, önceden geçerli olan veri yapıları kullanılarak sonradan tanımlanırlar.
 - Struct karmasik

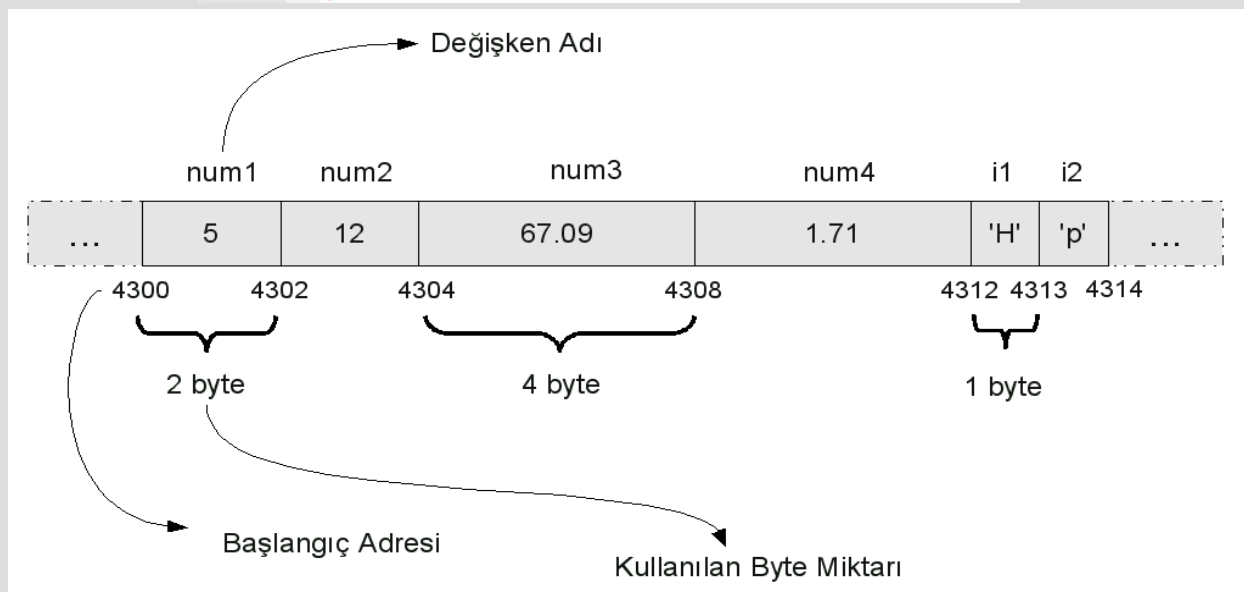
```
{
    float gercel;
    float sanal;
}
```
- Veri Yapıları Programlama Diline bağlıdır.

Veri Türleri

- Karakter (Character)
 - A, a, @, -, &, ?, >, <
- Tamsayı (Integer)
 - 1923, 1903, 10, -7
- Gerçek Sayı (Real Number) (Float - Double)
 - 3.14, 456.60
- Katar (String)
 - Beşiktaş, Türkiye, İnternet
- Dizi / Matris (Array / Matrix)
 - [12, 56, 67]

Verinin Bellekte Tutulması

```
1  #include<stdio.h>
2  int main( void )
3  {
4      // Değişkenler tanımlanıyor:
5      int num1, num2;
6      float num3, num4;
7      char i1, i2;
8      // Değişkenlere atama yapılıyor:
9      num1 = 5;
10     num2 = 12;
11     num3 = 67.09;
12     num4 = 1.71;
13     i1 = 'H';
14     i2 = 'p';
15
16     return 0;
17 }
```



Verinin bellekte tutulması

- Veri yapısı, ham olarak 1 ve 0'lardan oluşan verinin yorumlanmasını belirleyen biçimleme (formatting) düzenidir.
- Örneğin, 62 sayısı bilgisayar belleğinde üç farklı yöntemle tutulabilir;
 - Sayının ikili tabandaki karşılığı
 - İkili: 111110
 - Herbir karaktere özel kod verilmiş olan alfanümerik kod tablosu kullanılarak
 - ASCII karakter kümesine göre 00110110 00110010
 - BCD kodlama
 - BCD kodlamaya göre 0110 0010

Tamsayıların bellekte tutulması

- İkili
 - Sayının doğrudan ikili tabandaki karşılığı saklanır.
- 1'e tümleyen
 - 1'e tümlleme, bit dizisi içerisindeki 1'lerin 0, 0'ların 1 yapılmasıyla elde edilir.
- 2'ye tümleyen
 - 2'ye tümlleme, bit dizisi içerisindeki 1'lerin 0, 0'ların 1 yapıp sonuca 1 eklenmesiyle elde edilir. İkiye tümlleme, diğer bir deyişle, 1'e tümllemeye 1 eklemedir.
- BCD
 - Bir tamsayı, her bir rakamına ait BCD kodun kullanılmasıyla elde edilir. Örneğin 123 sayısı 3 haneli BCD bir tamsayıdır ve 0001 0010 0011 olarak kodlanır.

Tümleme işlemi ile çıkarma

- $(219)_{10} - (233)_{10}$
- İkili
 - $(219)_{10} = (11011011)_2$
 - $(233)_{10} = (11101001)_2$
 - $(-233)_{10} = (00010111)_2 \rightarrow$ İkiye göre tümleme
 $(11011011)_2 + (00010111)_2 = (11110010)_2$
 $(11110010)_2 = (-1110)_2 \rightarrow$ İkiye göre tümleme

İşaretli (Signed) ve İşaretsiz (Unsigned) Tam sayı

- İşaretli tam sayı hem negatif hemde pozitif tam sayıları kapsarken, işaretsiz tam sayı ise pozitif tam sayıları kapsar.
- Bir bitin işaret için ayrılması ile işaretli tam sayılar elde edilir. Örneğin,
 - 1110 ise işaretsiz olarak onlu tabanda 14 iken işaretli olarak -6'ya eşdeğerdir.

C# Değişkenler, kapladıkları alan (Byte), min ve max değerleri

Type	Size (byte)	Minimum Value	Maximum Value
sbyte	1	-128	127
short	2	-32,768	32,768
int	4	-2,147,483,647	2,147,483,647
long	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
byte	1	0	255
ushort	2	0	65,535
uint	4	0	4,294,967,295
ulong	8	0	18,446,744,073,709,551,615
float	4	1.5x10 power -45	3.4 x10 power 38
double	8	5.0x10 power -324	1.7x10 power 308

Karakter Veri Yapısı ve Kodlama (Encoding)

- Karakter, en temel veri yapılarından birisi olup tek tek karakterlerin veya art arda gelerek sözcüklerin, cümlelerin tutulduğu bir yapıdır.
- Kodlama (Encoding) işlemi bir karakter tablosuna göre yapılır; tablo değiştirildiğinde karakterlere karşı düşen kodlar değişeceğinden ham verinin içerdiği bilgi farklı görülür.
 - ASCII
 - EBCDIC
 - Unicode
 - UTF

ASCII (American Standard Code for Information Interchange)

- ASCII (Bilgi Değişimi İçin Amerikan Standart Kodlama Sistemi) Latin alfabesi üzerine kurulu 7 bitlik bir karakter setidir.
- İlk kez 1963 yılında ANSI tarafından standart olarak sunulmuştur.
- ASCII'de 33 tane basılmayan kontrol karakteri ve 95 tane basılan karakter bulunur.
- Kontrol karakterleri metnin akışını kontrol eden, ekranda çıkmayan karakterlerdir.

ASCII Tablosu

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Genişletilmiş ASCII

- Genişletilmiş ASCII karakterler ek karakter talebini karşılar. Genişletilmiş ASCII'de, ASCII'de yer alan (aşağıdaki grafikte görüntülenen 0–32 arası sayılar) 128 karaktere ek olarak, 128 karakter daha bulunur; böylece toplam karakter sayısı 256'ya ulaşır.
- Bu ek karakterlerle bile, birçok dilde 256 karaktere katılamayan simgeler vardır. Bu nedenle, bölgesel karakter ve simgeleri karşılamak için ASCII çeşitlemeleri vardır.
- Örneğin, Kuzey Amerika, Batı Avrupa, Avustralya ve Afrika dillerine ait yazılım programlarında [ISO 8859-1](#) olarak da bilinen ASCII tablosu kullanılır.

- EBCDIC (Extended Binary Coded Decimal Interchange Code)

- (Genişletilmiş İkilik kodlu Ondalık Değişim Kodu, okunuşu: ebsedik)
- IBM tarafından kullanılan bir karakter kümesi ailesidir. Harf, rakam, işaretleri karşılayan 256 farklı sembolü kodlayabilir.
- EBCDIC sisteminin değişik alt türleri vardır. Bu türler karşılıklı olarak uyum içinde değildir.
- IBM, kendine özgü olmak üzere türleri birbirine çeviren yazılımlar da üretmiştir. Örneğin; EBCDIC kodunda büyük A harfi 193 (C1hex), küçük a harfi ise 129 (81hex) sayısı ile kodlanmıştır.

Unicode

- Unicode (Evrensel Kod) Unicode Consortium organizasyonu tarafından geliştirilen ve her karaktere bir sayı değeri karşılığı atayan bir standarttır. Evrensel Karakter Seti (UCS) olarak bilinen ISO/IEC 10646 standardı ise, her iki organizasyonun işbirliği ile aynı sayısal karşılıkları taşımaktadır. Bu set,
 - Yeryüzündeki tüm karakterlere bir sayı değeri atamayı amaçlamaktadır.
 - Zaman içinde yeni karakterler eklenebilir ama eski karakterlerin sayı değerleri aynı kalır.
 - Sayı değerleri UCS-4 adlı 32 bitlik set üzerinden verilir. İlk 7 bit Group, sonraki 8 bit Plane, sonraki 8 bit Row, en son 8 bit Cell olarak gruplanır. İlk Plane (group = 0, plane = 0) Basic Multilingual Plane (BMP) olarak adlandırılır. BMP, UCS-2 adı verilen 16 bitlik sete karşılık gelmektedir.
 - UCS üzerindeki karakter kod noktaları genellikle u+0a31 biçiminde onaltılık sistemde sayılar olarak gösterilir.
 - u+0021 - u+007e arasındaki kodlar ASCII ile, u+00a0 - u+00ff arasındaki kodlar ISO-8859-9 ile aynı tutulmuştur.

Unicode Türkçe Karakterler

- Unicode kodlarından oluşan karakter dizilerini (metinleri) bilgisayarda verimli bir biçimde saklayabilmek amacıyla çeşitli karakter kodlamaları (encoding) geliştirilmiştir.
- Türkçe karakterlerin unicode karşılıkları şunlardır:
 - ğ - \u011f
 - Ğ - \u011e
 - ı - \u0131
 - İ - \u0130
 - ö - \u00f6
 - Ö - \u00d6
 - ü - \u00fc
 - Ü - \u00dc
 - ş - \u015f
 - Ş - \u015e
 - ç - \u00e7
 - Ç - \u00c7

Unicode Transformation Format (UTF)

- Bir Unicode dönüşüm biçimidir.
- Unicode karakterlerini değişken uzunluklu bayt gruplarıyla kodlamakta kullanılır.
 - UTF-1: UTF-8'in bir öncülüdür. ISO 2022 ile uyumluluğu artırır.
 - UTF-7: Bazen e-mail'lerde kodlamada kullanılan 7 bitlik bir dönüşüm biçimidir. Ancak, günümüzde bir internet standardı değildir.
 - UTF-8: 8 bitlik değişken genişlikte ASCII uyumluluğunu arttıran kodlama. *
 - UTF-EBCDIC: UTF-8'e benzer ama EBCDIC uyumluluğu için tasarlanmıştır.
 - UTF-16: 16-bit, değişken genişlik kodlama *
 - UTF-32: 32-bit, sabit genişlikli kodlama
- UTF-8 ve UTF-16 genellikle dönüşümde kullanılır.

UTF-8

- Rob Pike ve Ken Thompson tarafından geliştirilmiştir.
- UTF-8 kodlaması Unicode karakterlerini 1-6 byte uzunluğunda diziler olarak kodlar. ASCII kodlaması içinde 0-127 arasında kalan karakterler aynen kendi kodları ile kullanılır, diğerleri ise byte dizileri haline gelir.
 - Evrensel kodlama ile aynı sayfada farklı lisansları göstermek mümkün olabilmektedir.
 - Özel hazırlanmış klavyeler ile matematiksel ifadeleri yazmak da mümkündür.
 - Bütün bu farklı karakterler UTF-8 sayesinde aynı sayfada görüntülenebilmektedir.
 - Türkçe karakter desteği de vardır.
 - UTF kullanarak 1 milyondan fazla karakter kodlanılabilmektedir.
 - Sıklıkla kullanılan 65536 karakterin kodlaması bu 1 milyondan fazla kodlamada ilk sıralarda yer almaktadır.

ASCII Küçük büyük harf çevirme programı

- string - char
- int
- char ve int arasında Dönüşüm işlemleri
- Yazılan kodta istenen fonksiyonlar
 - Girilen bir kelimenin ascii karşılığının yazılması
 - Girilen bir kelimenin ikili karşılığının yazılması
 - Girilen bir sayının BCD karşılığının yazılması

Kesirli Sayı

- İkili sayı sistemi ve 10'lu sayı sistemine geçiş



$$1101.01_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

Labels in the diagram:

- "Basamak Değeri" (Digit Value) points to the '1' in 1×2^3 .
- "Basamak Ağırlığı" (Digit Weight) points to the 2^3 in 1×2^3 .
- "Taban Değeri" (Base Value) points to the '2' in 2^3 .

The final result is shown as $1101.01_2 = 13.25_{10}$.

$$S = \sum_{a=-\infty}^{\infty} B_a 2^a$$

Kesirli Sayı

- 10'lu sayı sisteminden ikili sayı sistemine geçiş

$$13.25_{10} = (?)_2$$

$$\frac{13}{2} = 6 + \text{kalan } 1$$

$$\frac{6}{2} = 3 + \text{kalan } 0$$

$$\frac{3}{2} = 1 + \text{kalan } 1$$

$$\frac{1}{2} = 0 + \text{kalan } 1$$

Buradan 1 1 0 1 elde edilir.

İkinci ve son kısımda ise kesirli kısmın dönüşümü yapılır.

$$0.25 \times 2 = 0.5 \text{ tam kısmı } 0$$

$$0.5 \times 2 = 1.0 \text{ tam kısmı } 1$$

Sonuç olarak 1 1 0 1 . 0 1 elde edilir.

$$13.25_{10} = 1101.01_2$$

Daha fazla örnek için:

http://data.obitet.net/obitet/bilgisayar/02_MikSis.pdf

Kayan Noktalı Sayı Formatı

- Kayan noktalı sayılar gerçel sayıların bilgisayar ortamındaki gösterim şekillerinden biridir.
- Gerçek dünyada sayılar sonsuza kadar giderken, bilgisayar ortamında bilgisayar donanımının getirdiği sınırlamalardan dolayı bütün sayıların gösterilmesi mümkün değildir. Bununla birlikte gerçekte sonsuza kadar giden birtakım değerler bilgisayar ortamında ortamın kapasitesine bağlı olarak yaklaşık değerlerle temsil edilirler.
- Bu sınırlamaların etkisini en aza indiren, sayıların maksimum miktarda ve gerçeğe en yakın şekilde temsilini sağlayan sisteme "Kayan-Noktalı Sayılar" sistemi denir.

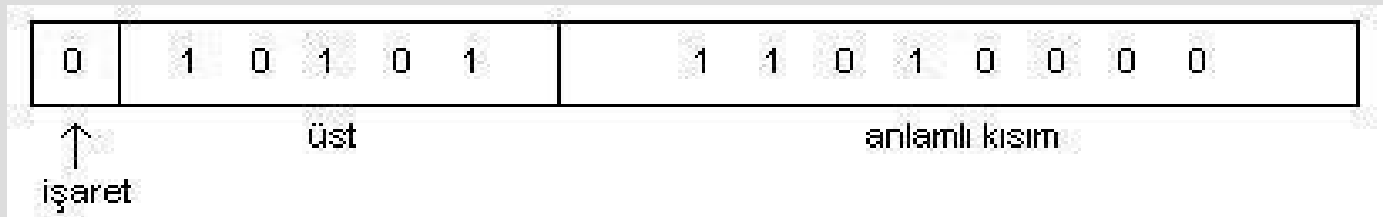
Kayan Noktalı Sayı Formatı

- Kayan noktalı sayılar, ikilik düzendeki sayıların bilimsel gösterimle gösterilmesidir. Kayan noktalı sayılar işaret, anlamlı kısım ve üst(2'nin üssü şeklinde) olmak üzere üç kısımdan oluşur.
 - $M \times B^E$
- Burada M kayan noktalı sayının mantis'i; B tabanı, E ise üssüdür. Kayan noktalı sayının gösteriminde anlamlı kısmın bitleri fazla olursa sayının duyarlılığı, üst bitleri fazla olursa gösterilebilecek sayı aralığı artar.
- Bu modele göre 13 sayısını gösterelim;

13 sayısı, $13 = 13,0 \times 100 = 1,3 \times 10^1 = 0,13 \times 10^2$ olarak ifade edilebilir.

İkilik sistemde düşünürsek: $13_{10} = 1101_2 \times 2^0 = 110,1_2 \times 2^1 = 11,01_2 \times 2^2 = 1,101_2 \times 2^3 = 0,1101 \times 2^4$

Son şekliyle ele alırsak, işaret biti 0, anlamlı kısım 11010000 ve üst 00101 olacaktır.



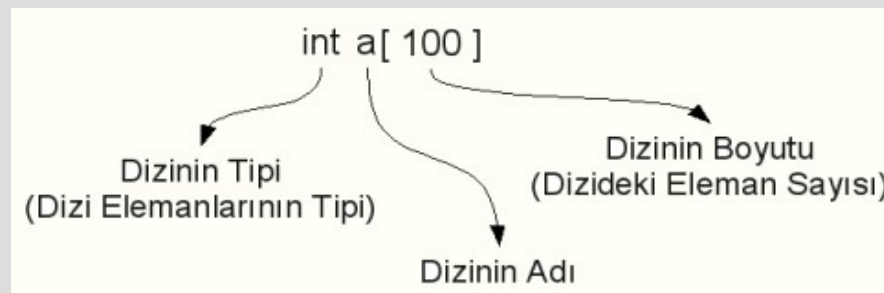
Daha geniş bilgi için: IEEE 754 standardı: Kayan noktalı sayıların her değişik aygıta uyumlu bir gösterime sahip olması için IEEE 754 standardı geliştirilmiştir.

Karakter Katarı / Sözlük (String)

- Bir katar, genellikle harflerden oluşan karakterler dizisidir.
 - C'de char tipi bir bilginin NULL karakteri ile sonlandırılmasıdır.
 - C# ve Java gibi dillerde string veri tipi kullanılmaktadır.
- C# string uygulamaları
 - Girilen bir string'in uzunluğunu bulan
 - Girilen bir string'i tersten yazan

Dizi / Matris

- Dizi bir kümedir. Aynı tipte verilere tek bir isimle erişmek için kullanılır.
- Bir dizinin bütün elemanları bellekte peş peşe saklanır.



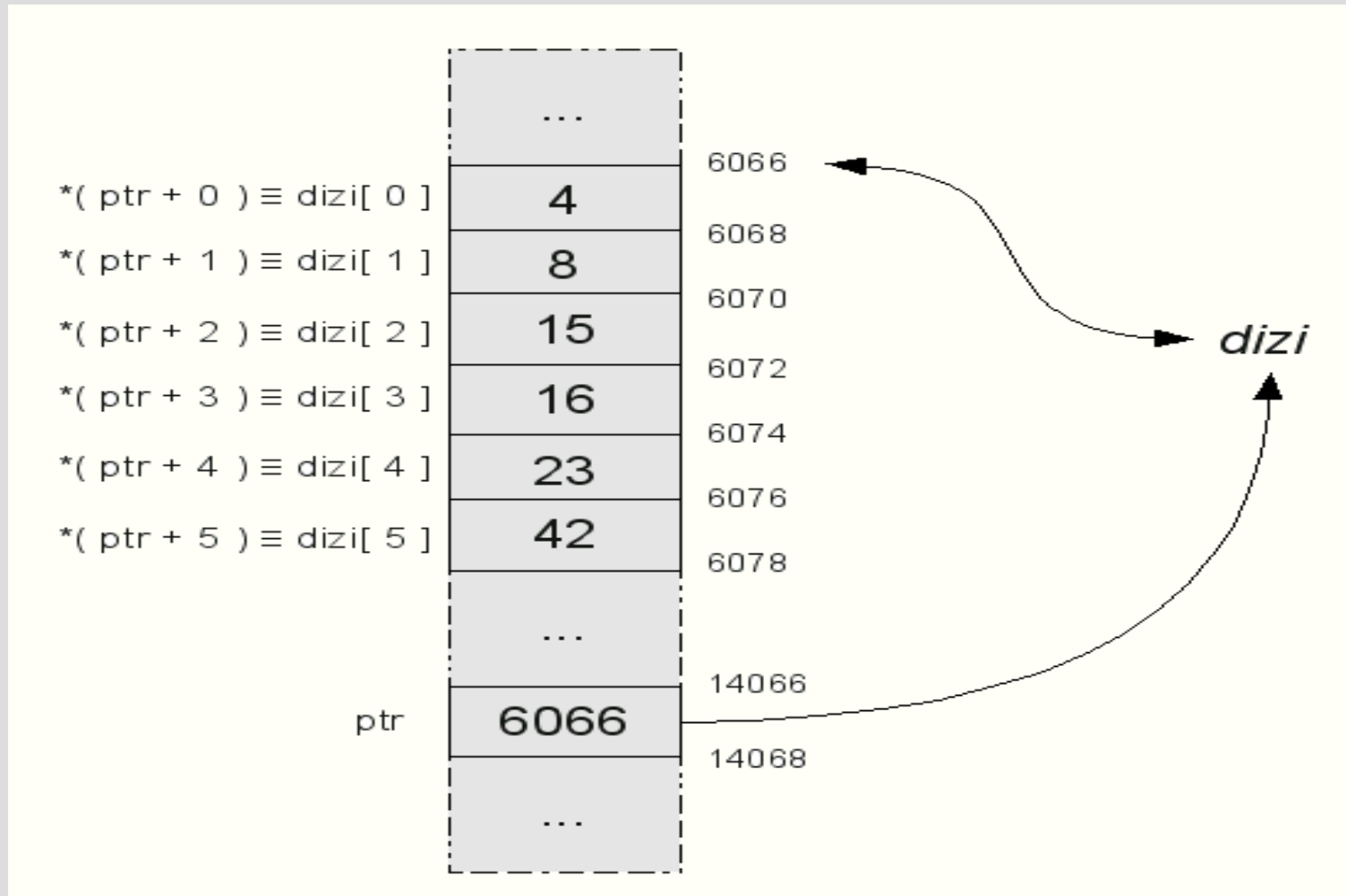
C# Dizi Tanımlama

```
int[] array = new int[5];
```

```
int[] array3;  
array3 = new int[] { 1, 3, 5, 7, 9 }; // OK  
//array3 = {1, 3, 5, 7, 9}; // Error
```

```
int[] array2 = { 1, 3, 5, 7, 9 };  
string[] weekDays2 = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };
```

Dizilere pointer ile erişim



- Daha fazla ayrıntı için C Pointer konusuna bakınız.

Çok Boyutlu Diziler

- Diziler, birden fazla boyut olabilir. Örneğin, aşağıdaki bildiriği, dört satır ve iki sütun iki boyutlu bir dizi oluşturur.
- Üç boyutlu bir oyun düşünüyorsanız en az üç boyutlu bir dizi üzerine çalışmalısınız. (X, Y, Z)

[illegible]

Diziler Soru

- Yandaki iki boyutlu matrisi inceleyip matristeki sayıları toplamını bulan programı yazın.
- Döngüyü kısaltmak mümkün müdür?

	A	B	C	D	E
A	0.00	0.71	5.66	3.61	4.24
B	0.71	0.00	4.95	2.92	3.54
C	5.66	4.95	0.00	2.24	1.41
D	3.61	2.92	2.24	0.00	1.00
E	4.24	3.54	1.141	1.00	0.00

Tanımlamalı Veri Yapıları

- Tanımlamalı veri yapısı, temel veya daha önceden tanımlanmış veri yapılarının kullanılıp yeni veri yapıları oluşturulmasıdır. Temel veri yapıları, genel olarak tüm uygulamalarda gereksinim duyulan ve bolca kullanılan bir olgudur.
 - Struct veya Class
 - Ortaklık (Union)

Struct veya Class

- Birden çok veri yapısının bir araya getirilip yeni bir veri yapısı, bir aile ortaya çıkarmaktır.
- Struct tanımlanması C, C# veya Java dillerinde struct deyiimiyle yapılır.
- C# ve Java dillerindeki Sınıf (Class) kavramına benzer. (Farklılıklar için geçen seneki Nesneye Dayalı Programlama dersi notlarına bak.)

```
struct ogrenci{  
    char  ad[10],soyad[20];  
    long  no;  
    short sinif;  
};
```

Union

- Bir programda veya fonksiyonda değişkenlerin aynı bellek alanını paylaşması için ortaklık bildirimi union deyimi ile yapılır. Bu da belleğin daha verimli kullanılmasına imkan verir. Bu tipte bildirim yapılırken struct yerine union yazılır.

```
union paylas{  
    float f;  
    int    i;  
    char   kr;  
};
```