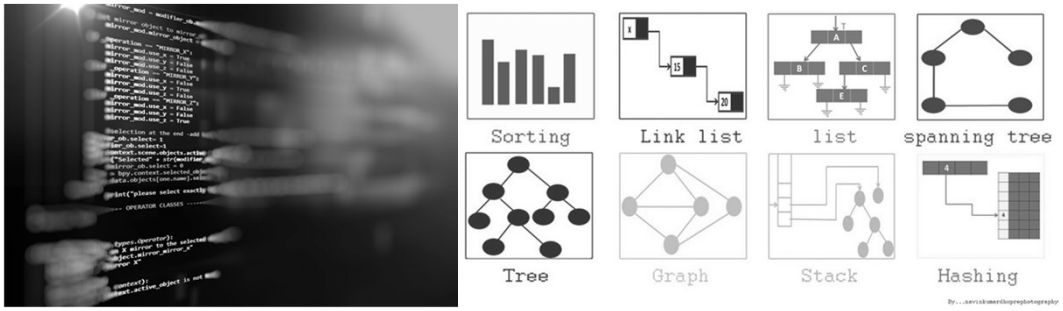
Veri Yapıları



Hafta 2

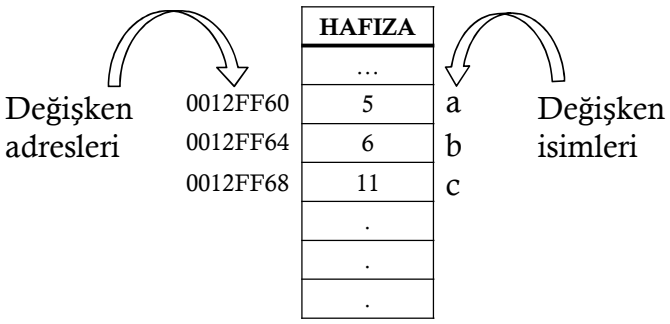
HATIRLATMA !

```c
#include <stdio.h>
int main()
{/* sizeof() */
  printf("hafızada %d byte yer tutar (char)\n",sizeof(char));
  printf("hafızada %d byte yer tutar (short int)\n",sizeof(short int));
  printf("hafızada %d byte yer tutar (int)\n",sizeof(int));
  printf("hafızada %d byte yer tutar (long)\n",sizeof(long));
  printf("hafızada %d byte yer tutar (float)\n",sizeof(float));
  printf("hafızada %d byte yer tutar (double)\n",sizeof(double));
  return 0;
}
```

```
hafızada 1 byte yer tutar (char)
hafızada 2 byte yer tutar (short int)
hafızada 4 byte yer tutar (int)
hafızada 8 byte yer tutar (long)
hafızada 4 byte yer tutar (float)
hafızada 8 byte yer tutar (double)
```

HATIRLATMA ! (Hafıza(Bellek) Kavramı)

```
/* iki sayının toplamı*/
#include <stdio.h>
int main()
{
    int a,b,c;
    a=5;
    b=6;
    c=a+b;
    printf("%d", c);
    return 0;
}
```



HATIRLATMA ! (ASCII)

- ASCII (American Standard Code for Information Interchange)
- Bilgi Değişimi İçin Amerikan Standart Kodlama Sistemi (7bitlik) → Her karakter için 7bit:

- 65-90 (10luk sistem)--> A-Z
- 97-122 (10luk sistem)--> a-z

İkilik	Sekizlik	Onluk	On altılık	Karakter
100 0001	101	65	41	A

- Günümüzde Extended ASCII kullanılmakta olup karakterler 8bit (1byte) yer tutar. Metin dosyalarının boyutu yaklaşık olarak içerdiği karakter sayısı kadardır.

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SoH	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	SoTxt	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	EoTxt	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EoT	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enq	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Ack	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Bsp	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	HTab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LFeed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VTab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FFeed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SOut	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SIn	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAck	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Syn	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	EoTB	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Can	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EoM	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Sub	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Esc	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FSep	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GSep	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RSep	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	USep	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Delete

charstable.com

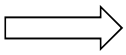
HATIRLATMA ! (ASCII)

```c
/* ascii */
#include <stdio.h>
int main()
{
  char kar;
  printf("Bir karakter giriniz: ");
  scanf("%c",&kar);
  printf("\nKarakter:%c ",kar);
  printf("\nOnlukSis:%d ",kar);
  printf("\n16likSis:%x ",kar);
  printf("\n8likSis :%o ",kar);
  printf("\nAdresi  :%p ",kar);
  return 0;
}
```

```c
/* ascii */
#include <stdio.h>
int main()
{
char kar;
printf("Bir karakter giriniz: ");
scanf("%c",&kar);
printf("\nGirdiğiniz Karakter:%c ve bir
sonraki karakter: %c ",kar, kar+1);
return 0;
}
```

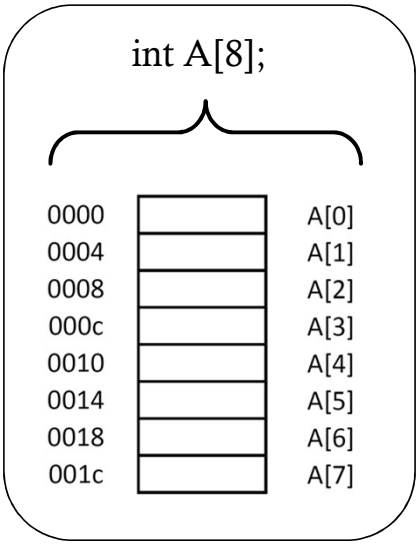
C/C++

```c
// Program Şablonumuz C
#include <stdio.h>


void yaz();

int main()
{
   int sayi;
   yaz();

   scanf("%d",&sayi);
   printf("%d",sayi);

   return 0;
}

void yaz()
{
   printf("Veri Yapilari\n");
}
```



```c
// Program Şablonumuz C++
#include <iostream>
using namespace std;

void yaz(); //fonksiyon prototipi

int main()
{
   int sayi;
   yaz(); //fonksiyon çağrımı

   cin>>sayi;
   cout<<sayi;

   return 0;
}

void yaz() //fonksiyon tanımı
{
   cout << "Veri Yapilari"<<endl;
}
```

Diziler (Arrays)



Dizi veri yapısında, diziye ait olan veriler bellekte art arda tutulurlar.

Dizinin başlangıç adresi veya adı bilindiğinde herhangi bir elemana kolayca erişilebilir.

```
//dizi boyutunu bulalım

#include <iostream>
using namespace std;

int main()
{
int dizi[8];

cout<<sizeof(dizi);
return 0;
}
```

```
//istenilen elamana ulaşma

#include <iostream>
using namespace std;

int main()
{
   int dizi[8];
   dizi[4]=43;
   cout<<dizi[4];
   return 0;
}
```

Örnek : diziyi fonksiyona parametre olarak yollayalım:

```
//diziyi fonksiyona yollama işlemi
#include <iostream>

using namespace std;

int fonk(int[],int);

int main()
{
   int sonuc;
   int dizi[5]={3,4,7,1,9};

   sonuc=fonk(dizi,5);
   cout<<sonuc;

   return 0;
}
```

```
//DEVAMI

int fonk(int d[],int boy)
{
   int toplam=0;
   for(int i=0;i<boy;i++)
   {
      toplam+=d[i];
   }

   return toplam;
}
```

Diziler – Karakter Dizileri

```
//bir kelimeyi diğerine ekleyelim
#include<iostream>

using namespace std;

void ekle(char dizi1[], char dizi2[]);

int main()
{
char dizi1[20];
char dizi2[20];
cout<<"Bir seyler yazin : ";
cin>>dizi1; //kelimeyi okur

    cout<<"eklenecek diziyi giriniz : ";
    cin>>dizi2;

ekle(dizi1, dizi2);
return 0;
}
```

```
//DEVAMI

void ekle(char dizi1[], char dizi2[])
{
  int i,j;
  for (i = 0; dizi1[i] != '\0'; i++)  ;
  for (j = 0; dizi2[j] != '\0'; j++)
    dizi1[i+j] = dizi2[j];
  dizi1[i+j] = '\0';
  cout<<"Sonuc : "<< dizi1<<endl<<endl;
}
```

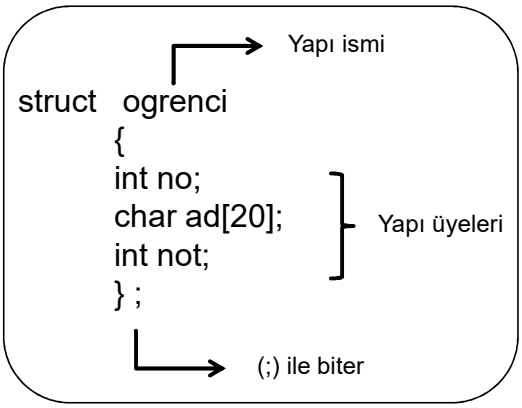
Diziler – Karakter Dizileri

```
/* Bir katardan istenilen bir karakteri çıkaralım,
Hatırlarsanız ödevdi */

#include <iostream>

using namespace std;
int main()
{
  char dizi[]="denemeler";
  char cikar;
  int i = 0;

  cout<< "Cikarmak istediginiz karakteri giriniz: ";
  cin>> cikar;
```

```
//DEVAMI

  while(dizi[i]!='\0')
  {
    if(dizi[i]==cikar)

      for(int j=i;j<10;j++)

        dizi[j]=dizi[j+1];
    else
    i++;
  }
  cout<<dizi;

  return 0;
}
```

Yapılar (Structures)

Temel veri yapılarının birleştirilmesi ile oluşturulmuş tanımlamalı bir veri yapısıdır.



ogrenci yapısının büyüklüğü :

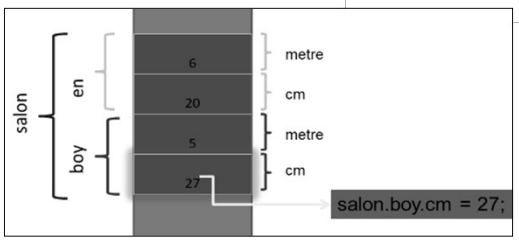
⇩

4(int)*1 + 1(char)*20 + 4(int)*1 = 28 byte

Yapılar (Structures)

```
#include <iostream>
using namespace std;

struct mesafe   //yapı tanımlaması
  {
  int metre;
  float cm;
  };

struct oda  //yapı tanımlaması
  {
  mesafe en;
  mesafe boy;
  };
```

```
 //DEVAMI
int main()
  {
  oda salon; //değişken tanımlaması
  salon.en.metre=6;
  salon.en.cm=20;
  salon.boy.metre=5;
  salon.boy.cm=27;

  float u=salon.en.metre + salon.en.cm/100;
  float g=salon.boy.metre + salon.boy.cm/100;

  cout << " Salon Alanı ..: " << u*g
<< " metre kare dir. "<<endl;
return 0;
}
```



Output :

```
Salon alanı..: 32.674 metre kare dir.
```

Yapılar (Structures)

```
//Fonksiyona struct gönderme işlemi

#include "iostream"

using namespace std;
struct kare
{
  double a;
};
double alan(struct kare r)
{
  return r.a*r.a;
}
```

```
//DEVAMI

int main()
{
  struct kare kare1;
  cout<<endl;

  cout<<"karenin kenar uzunlugunu giriniz: "<<endl;
  cin>>kare1.a;

  cout << "Karenin alani: "<<alan(kare1) << endl;

return 0;
}
```

```
karenin kenar uzunlugunu giriniz:
9
Karenin alani: 81
```