BMM 111
Bilgisayar Programlama-I

Dr. Öğr. Üyesi Mustafa İSTANBULLU

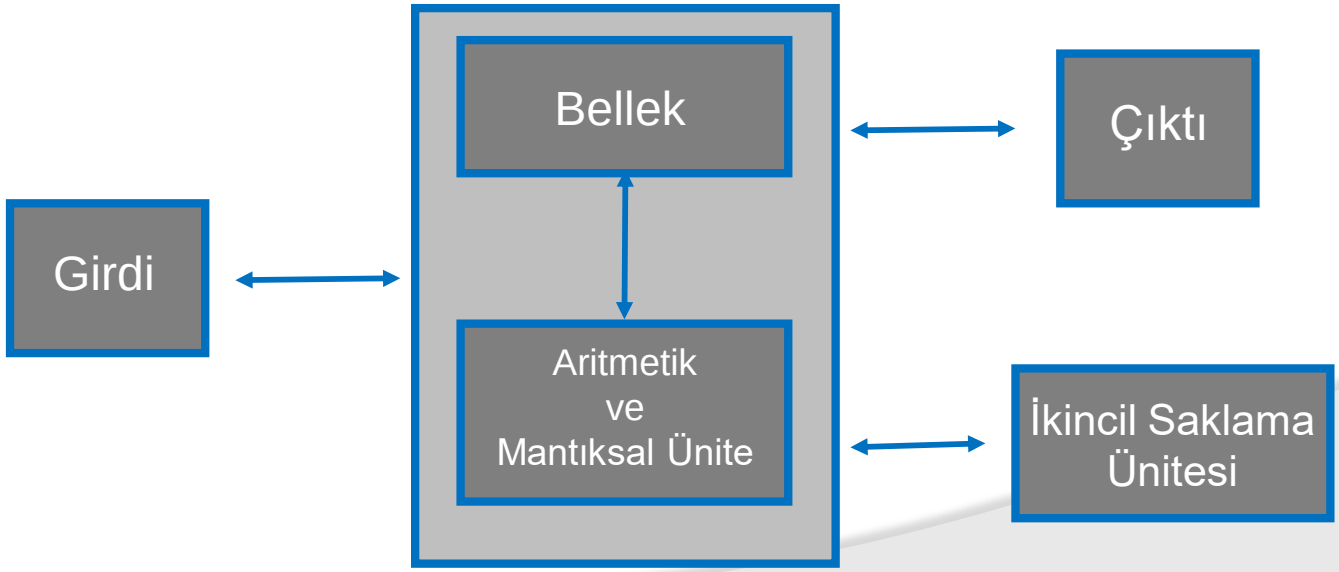
Çukurova Üniversitesi
Mühendislik Fakültesi
Biyomedikal Müh. Böl.

E-mail:mm.istanbullu@gmail.com

Not: Slaytlar, kaynakça bölümünde verilen listeden faydalanılarak hazırlanmıştır.

1. Bölüm Genel Kavramlar

- 1.1. Bilgisayarın Temel Birimleri
- Günümüzdeki bilgisayar sistemleri, elektronik devreler aracılığı ile bilgileri saklayabilme ve işleyebilme yeteneğine sahiptirler.
- Bilgisayarların temel birimleri Şekil 1.1'de gösterilmektedir.



Şekil 1.1. Bilgisayarın Temel Birimleri.

1. Bölüm Genel Kavramlar

Bilgisayarlar bilgileri girdi (input) olarak kendi bünyesine alma, bunları işleme ve sonuçları çıktı (output) olarak çeşitli şekillerde sunma yeteneğine sahiptirler.

Günümüzde kullanılan girdi ünitelerine klavye (keyboard) ve fare (mouse) örnek olarak verilebilir.

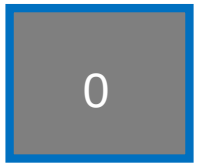
Çıktı birimleri ise, C programı aracılığı ile işlenen bilginin diğer bir bilgisayar, ekran ya da yazıcı gibi farklı birimlere aktarılmasını sağlar.

1. Bölüm Genel Kavramlar

- Bilgisayar, bilgileri işlemek amacıyla, Merkezi İşlemci Birimi (central processing unit-CPU)'ni kullanır.
- Bilgisayar bilgileri iki farklı şekilde saklar:
- Bunlardan birisi bellek (memory) olarak adlandırılan kısımdır.
- Bellek hücrelerden oluşur ve bilgilerin kısa süreli olarak saklanmasını sağlar.
- Bellek içinde saklanmış olan bilgi, CPU içinde bulunan Aritmetik ve Mantıksal Birim (arithmetic and logic unit-ALU) yardımıyla işlenir.
- Matematiksel işlemler ve mantıksal karşılaştırmalar gibi tüm işlemler burada yapılır.
- Bilgi daha sonra uzun süreli olarak saklanmak istenirse bu amaçla ikincil saklama (secondary storage) üniteleri olarak da adlandırılan disk, CD, flash disk, HD gibi araçlarda kullanılır.

1.2. Bilgi Saklama

- Bilgisayarda bilgi saklamaya yarayan en küçük elektronik bilgi alanı b i t olarak adlandırılır.
- Bir bit, aşağıdaki şekilde gösterildiği gibi içerisinde 1 yada 0 değeri taşıyabilen elektronik bilgi saklama alanıdır.



Sıfır değeri içeren bir bit

Bir değeri içeren bir bit

Şekil 1.2. Bir bit bilgi saklama.

- Veri saklarken, sekiz bitin yanyana getirilerek kullanımı çok yaygın bir hale gelmiştir ve bu nedenle sekiz bitlik veri gruplarına b y t e adı verilmiştir.
- Aşağıdaki şekilde görüldüğü gibi, bir byte'lık bilgi alanı, içindeki her bir bitlik bilgi alanına farklı değerler atayarak, değişik bilgileri saklayabilir.

Şekil 1.3. Bir byte bilgi saklama.

1. Bölüm Genel Kavramlar

- Bir byte alanı içinde 256 (2^8) farklı bilginin saklanması mümkündür.
- Günümüzde bilgisayar sistemlerinde; ses, görüntü, hareketli görüntü, sayısal değerler, alfasayısal değerler gibi farklı bir çok yapıda değerin saklanması ve işlenmesi mümkün olabilmektedir.
- Bundan sonraki kısımda, sayısal ve alfasayısal değerlerin nasıl saklandığı ile ilgili kısa bilgiler verilecektir.

1.2.1. Sayısal Değerler

- Sadece sıfır (0) ve bir (1) değerlerini kullanarak sayısal değerlerin bilgisayarlarda tanımlanmasını sağlamak amacıyla ikili sayı sistemi kullanılmaktadır.
- Yani bu değerler iki tabanına dönüştürülerek bilgisayarda saklanmaktadır.
- İkili sayı sistemini daha iyi anlamak için, öncelikle günlük yaşantımızda çok yoğun olarak kullandığımız onluk sayı sistemine bir göz atalım.
- Onluk sayı sisteminde örneğin 543 sayısının açılımını aşağıdaki gibi yapmak mümkündür.

$$(543)_{10} = (5\times10^{2}) + (4\times10^{1}) + (3\times10^{0})$$
$$(543)_{10} = (5\times100) + (4\times10) + (3\times1)$$

- Şimdi aynı durumu ikili sayı sisteminde bir örnekte inceleyelim:

$$(101)_2 = (1\times2^2) + (0\times2^1) + (1\times2^0)$$

Buradan;

$$(101)_2 = (1\times4) + (0\times2) + (1\times1) = (5)_{10}$$

Böylece onluk tabana göre 5 olan bir sayının değeri, ikili tabana göre $(101)_2$ ile göstermek mümkündür. Bu değeri bir byte içinde aşağıdaki gibi gösterebiliriz:

$$(00000101)_2$$

- Bu durumu genellersek, ikili sayı sistemindeki sayıları onluk sayı sistemindeki karşılıkları yandaki tabloda gösterildiği gibi bulunabilir.
- Bu tabloda sadece 4 bitlik bir gösterim kullanılmıştır. Sayısal değerler, bilgisayar sisteminde ikili sayı sistemi kullanılarak gösterilebilmektedir.
- Peki alfasayısal (alphanumeric) değerleri bu sistemde nasıl gösterebiliriz?

ikili	onluk
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
…….	…

1.2.2. Alfasayısal Değerler ve Özel Karakterler

- Alfasayısal (alphanumeric) değerler, karakterlerin ve sayısal değerlerin tamamını kapsayan değerler kümesidir.
- Alfasayısal değerler; 1, 2 gibi sayısal değerleri içerebileceği gibi; A, B gibi alfabetik karakterleri de kapsayabilmektedir.
- Alfasayısal değerlerin ve noktalama işaretleri gibi diğer özel karakterlerin bilgisayar sisteminde saklanması ve kullanılması amacıyla her bir değere karşılık gelen bir sayısal değer atanmıştır.
- Bu değerler, bilgi değiş tokuşu için Amerikan kod standardı olarak isimlendirebileceğimiz ASCII (American Standard Code for Information Interchange) kod tablosunda tanımlanmıştır.

1. Bölüm Genel Kavramlar

- ASCII kodlama sistemi her sembol için 8 bit (1 byte) kullanmaktadır.
- Sekiz bit kullanarak 0 ila 255 rakamları ile toplam 256 adet sembol temsil edilebilmektedir.
- Tablo 1 ve Tablo 2' de ASCII kodlar ve karşılığı karakterler verilmiştir.
- Onluk sistemdeki (Decimal) ASCII kodlar ve karakter karşılıkları bu tablolarda görülmektedir.

Tablo 1. Standart ASCII kod tablosu.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

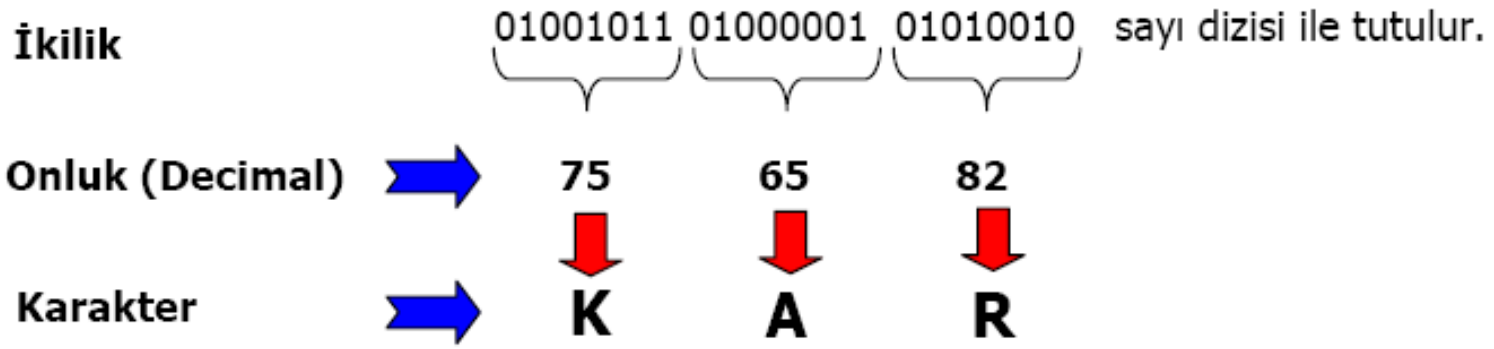
Source: www.asciitable.com

Tablo 2. Genişletilmiş ASCII kod tablosu.

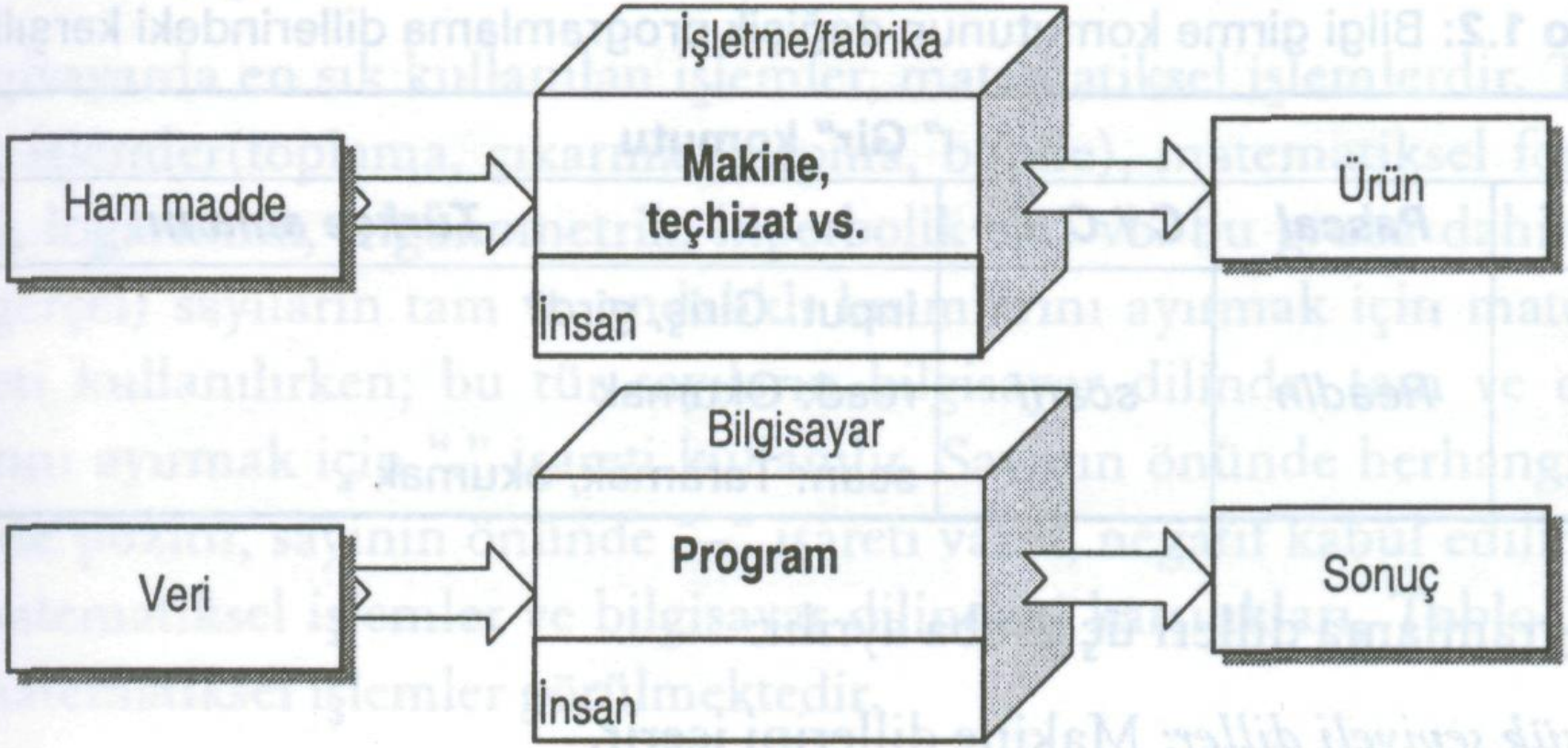
128	Ç	144	É	160	á	176	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	│	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	ÿ	168	¿	184	╕	201	╔	217	┘	233	Θ	249	∙
137	ë	153	Ö	169	⌐	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	█	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	ⁿ
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	φ	253	²
141	ì	158	₧	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

Source: www.asciitable.com

- Örneğin KAR kelimesi 75 , 65 ve 82 nolu ASCII karakterlerdir ve bilgisayarda ikili sayı sistemi karşılığı olan



1.2.3 Program ve programlama dili nedir?



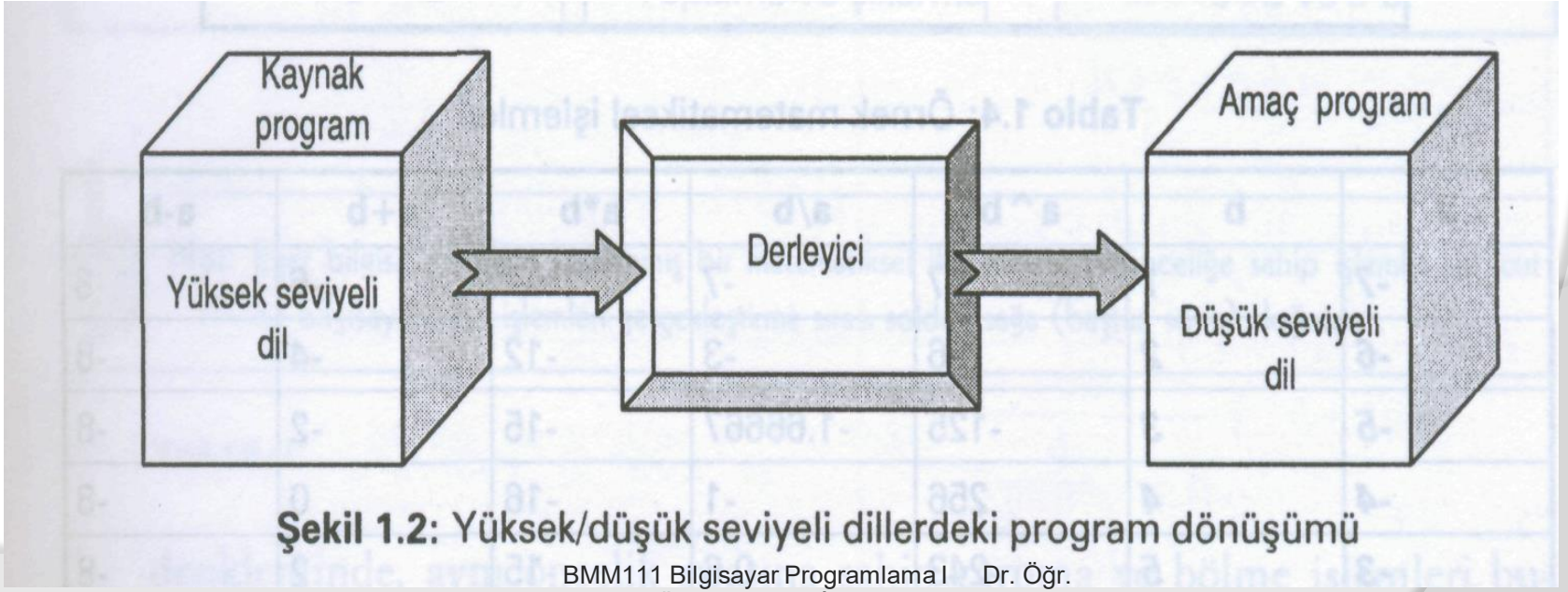
Şekil 1.1: İşletme/fabrika ve bilgisayardaki işlem akışı

1.2.3 Program ve programlama dili nedir?

- İnsanla bilgisayar arasındaki iletişim aracı
- Programlama dili, programcı ile bilgisayar arasındaki iletişimi sağlayan bir araç olup programların yazılımında kullanılan bir notasyondur (simgeler ve özel komutlar, komut parçacıkları).

1.2.4 Program dönüşümü

- Düşük seviyeli diller (assembly : MOV AL, 61h)
- Orta seviyeli diller (PIC programlama: SUBWF f,d [d = f − W])
- Yüksek seviyeli diller (C/C++, Pascal, QBasic : printf(), writeln(), PRINT)



Şekil 1.2: Yüksek/düşük seviyeli dillerdeki program dönüşümü

1. Bölüm Genel Kavramlar

Bilgisayarda problem çözme aşamaları:

- Problemin tanımı
- Çözüm yolunun tespiti
- Algoritmanın hazırlanması
- Akış diyagramının çizilmesi
- Programın hazırlanması
- Hazırlanan programın test edilmesi
- Uygulama

1.3 Bilgisayara İstediğimiz İşlemleri Nasıл Yaptırırız?

- Bilgisayarlar, kendilerine bir komut (emir) verilmediği sürece herhangi bir işlemi gerçekleştirebilme yeteneğine sahip değildirler.
- Bilgisayara, bir işin nasıl yapılması gerektiği ile ilgili tüm ayrıntıların detaylı ve düzenli bir biçimde aktarılması gerekir.
- İşte bilgisayarlara bu ayrıntıları aktarma işlemi sırasında bir işin nasıl yapılması gerektiği algoritmalar (algorithms) ile tanımlarız.
- Bir algoritmayı akış şemaları (flow charts) yardımı ile görsel olarak düzenleyebiliriz.

1. Bölüm Genel Kavramlar

ALGORİTMA NEDİR? NE İŞE YARAR?

- Algoritma aslında bilgisayarımıza yapmasını istediğimiz işi hangi adımları takip ederek gerçekleştirmesi gerektiğini anlatan adımlardır.

- Bu adımlar daha sonra program komutlarına dönüştürülecek ve bilgisayarımızın anlayabileceği bir dil ile bilgisayara aktarılacaktır.

1.3.1 Algoritma

- Algoritma, günlük hayatımızda yaptığımız işler sırasında sıklıkla kullandığımız bir yöntemdir.

- Algoritma, bir işin gerçekleştirilmesi aşamasındaki olası bütün adımları tek tek tanımlayan adımlar zinciridir.

Örneğin, bir arkadaşımıza telefon ederken kullandığımız algoritma aşağıdaki gibi oluşturulabilir:

1. B a ş l a
2. Ahizeyi kaldır
3. Çevir sinyalini kontrol et, sinyal yoksa arızaya haber ver ve 9. adıma git
4. Eğer telefon numarası yurt dışında ise iki kere sıfır tuşuna bas ve ülke ve alan kodunu tuşla ve 6. adıma git
5. Eğer telefon numarası şehir dışında ise bir kere sıfır tuşuna bas ve alan kodunu tuşla
6. Telefon numarasını tuşla
7. Eğer hat meşgul ise ya da cevap vermiyorsa 9. adıma git
8. Telefon konuşmasını gerçekleştir
9. Telefonu kapat
10. B i t i ş.

1. Bölüm Genel Kavramlar

- Bir bilgisayar programı hazırlanmadan önce mutlaka algoritması hazırlanmalı ve bu algoritma üzerinde en etkin yöntem denenerek bulunmalıdır.
- Böylelikle yazılacak olan bilgisayar programının adımları önceden belirlenmiş olur ve daha güvenli, doğru ve etkin çalışması sağlanır.
- Algoritması hazırlanan problem çözümü daha sonra algoritmada belirlenen adımlara uygun bir şekilde herhangi bir programlama dili (C, Cobol, Pascal, Fortran gibi) aracılığı ile bilgisayarın anlayacağı şekle dönüştürülür.

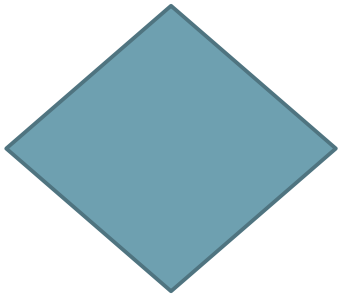
1.4.1 Akış Şeması

- Akış şemaları, algoritmalarda verilen her adımın görsel olarak anlatılması amacıyla kullanılan yöntemlerden birisidir.
- Akış şemalarında yaygın olarak kullanılan sembollerden bazıları:

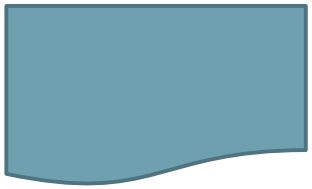
Başla/Bitir Sembolü: Başlangıç ve bitiş sembolü olup, akış şemasının başlangıç ve bitiş noktaları bu gösterim kullanılarak belirlenir.

Giriş sembolü: Bilgisayara bilgi girişini ifade etmektedir.

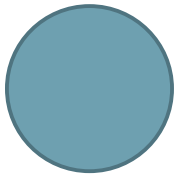
İşlem sembolü: Hesaplamalar veya matematiksel işlemler için kullanılan semboldür.



Koşul (karar) sembolü: Bazı kararlara bağlı olarak farklı yolların izlenebileceği durumlarda bu sembol kullanılır. Bu şekil içinde tanımlanan koşula veya soruya göre farklı kararların alınması sağlanır.



Doküman sembolü: Program sırasında çıktı olarak gösterilmek istenen veriler ve işlemler bu şekil ile tanımlanır.



Bağlantı (bağlama) sembolü: Akış şemasının çeşitli bölgelerini aynı sayı veya harfle birleştirerek akış şemasındaki kopuklukları bağlar.

Akış sembolü: Ok yönü akış yönünü göstermek üzere diğer akış şeması sembollerini bağlar.

Algoritmada Kullanılan Terimler

- Tanımlayıcı (programdaki değişkenleri, sabitleri, kayıt alanlarını, özel bilgi tiplerinin adlandırılması veya belirlenmesi)
- Değişken (x, ad, tel_no, sayi1 vs.)
- Aktarma (değişken = ifade)
- Sayaç (sayac = sayac + 1, x = x+3, s=s-5)

Algoritmada Kullanılan Terimler

- Döngü
  - Döngü değişkeninin başlangıç ve bitiş değeri,
  - Artma ya da azalma miktarı belirlenir.
- Ardışık toplama ve çarpma
  - top_degis= top_degis+sayi
  - carp_degis= carp_degis*sayi

Örnek: Bir sınıfta 60 öğrenci vardır. Herhangi bir A dersi için sınıfın not ortalamasının (aritmetik ortalama) bulunması istenmektedir.

a) Not ortalamasını bulan algoritmanın adımlarını yazınız.
b) Bu algoritmaya karşı düşen akış diyagramını çiziniz.

Çözüm: Algoritmada öğrenci sayısı maksimum 60 olup N değişkeni ile öğrenci sayısını saydıralım.

NOT değişkeni ile öğrencilerin notu,

TOPLAM değişkeni ile notların toplamı,

ORTALAMA değişkeni ile not ortalaması gösterilsin.

Algoritmanın Adımları

1. İlk olarak not toplamı (TOPLAM değişkeni) sıfıra ve öğrenci adedi (N değişkeni) 1'e eşitlenir.

   TOPLAM = 0

   N=1

2. Bir öğrencinin notu okunur. NOT değişkenine aktarılır.

3. Okunan not, not toplamına ilave edilir.

   TOPLAM = TOPLAM + NOT

4. Öğrenci adedi bir arttırılır.

   N=N+1

Algoritmanın Adımları

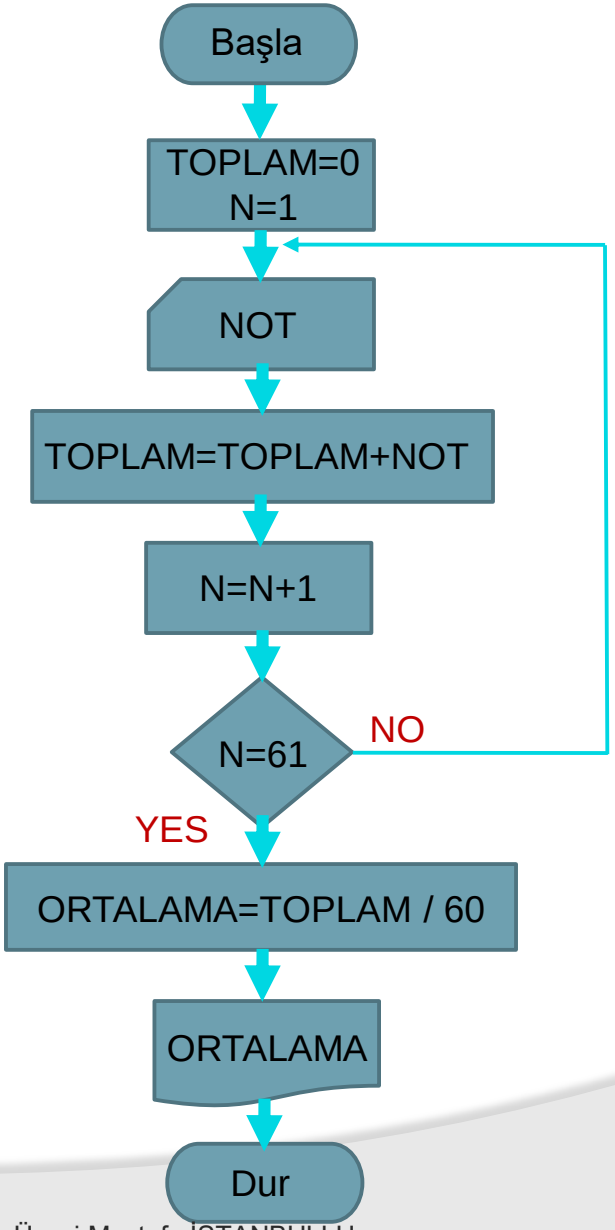
5. Öğrenci adedi 60'a eşit veya küçük ise ikinci adıma, eğer büyük ise bir sonraki adıma (6.) gidilir.

6. Bu adımda not ortalaması hesaplanır.

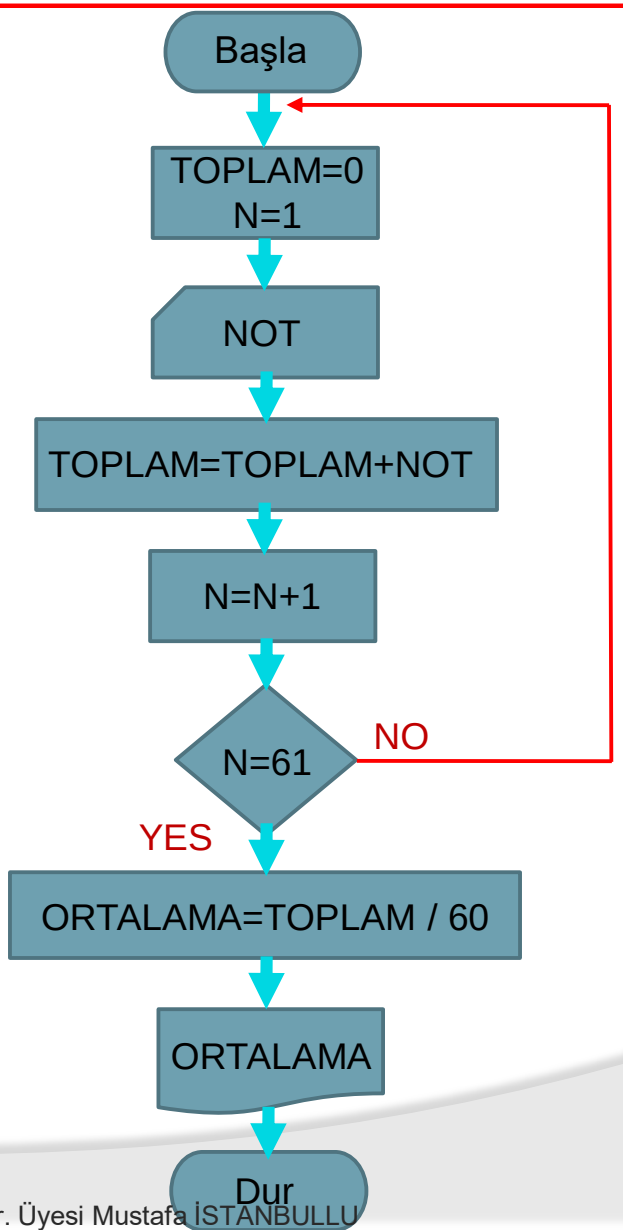
ORTALAMA = TOPLAM / 60

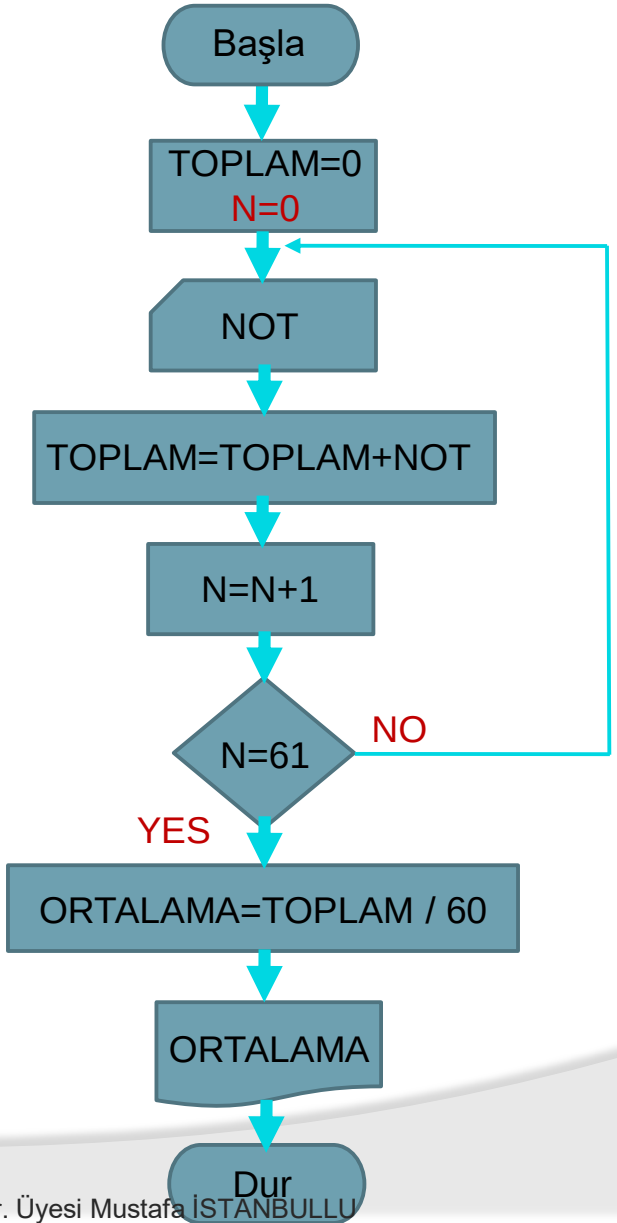
7. ORTALAMA ekrana yazdırılır.

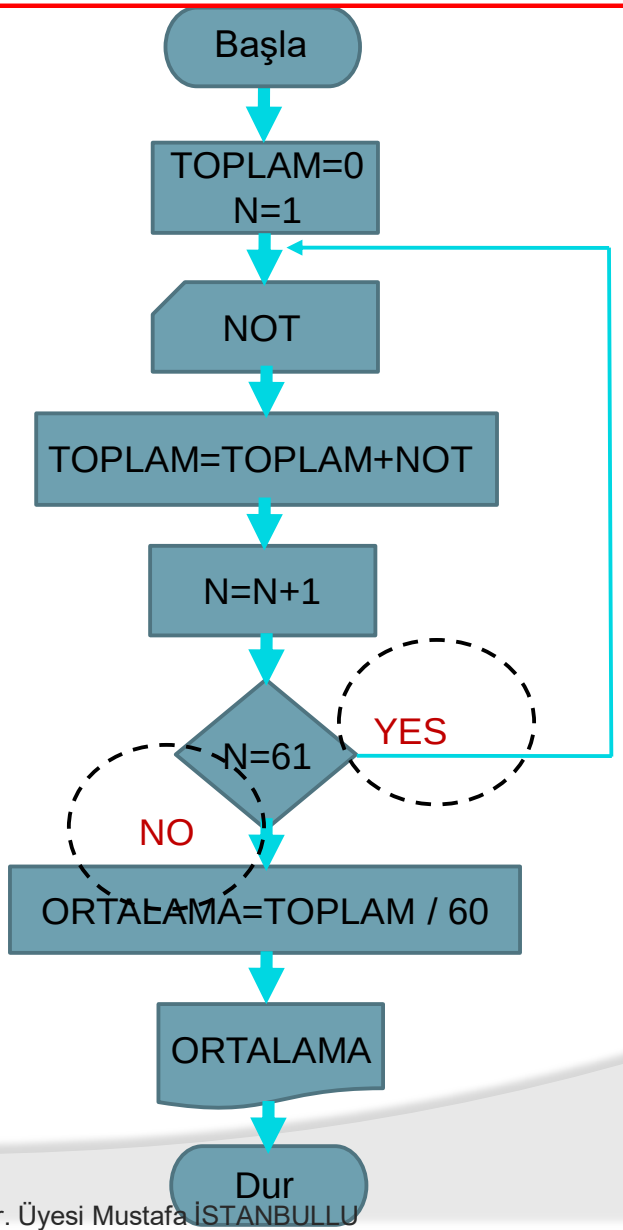
8. Yapılan işlem sonlandırılır.

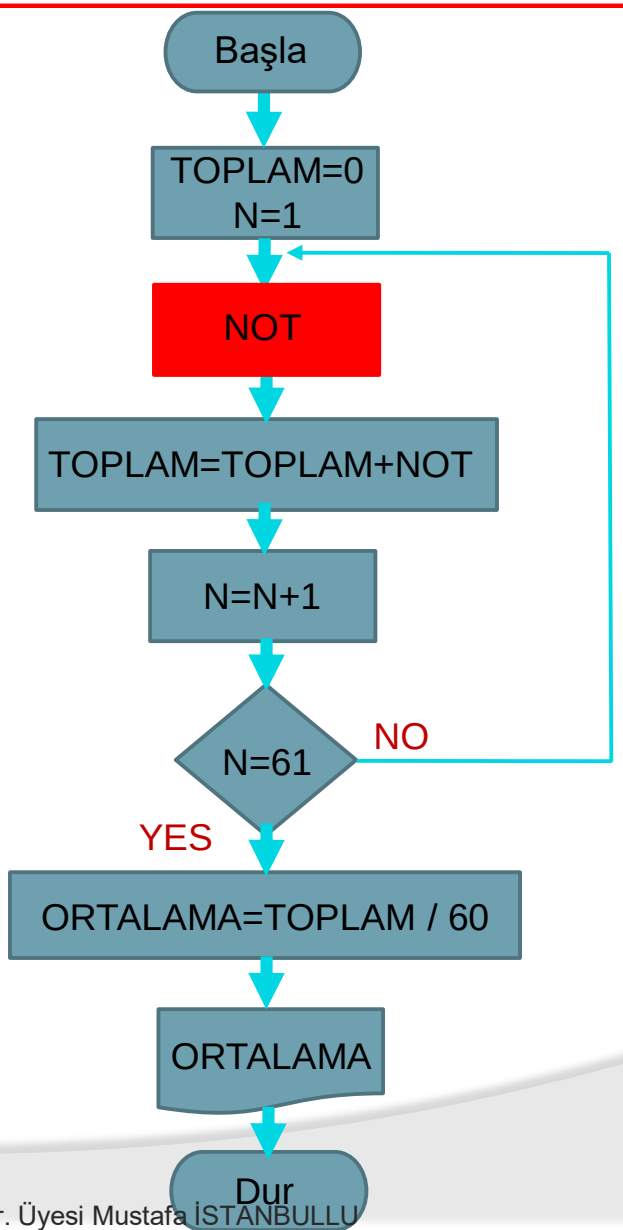

Başla
TOPLAM=0
N=1
NOT
TOPLAM=TOPLAM+NOT
N=N+1
N=61
NO
YES
ORTALAMA=TOPLAM / 60
ORTALAMA
Dur

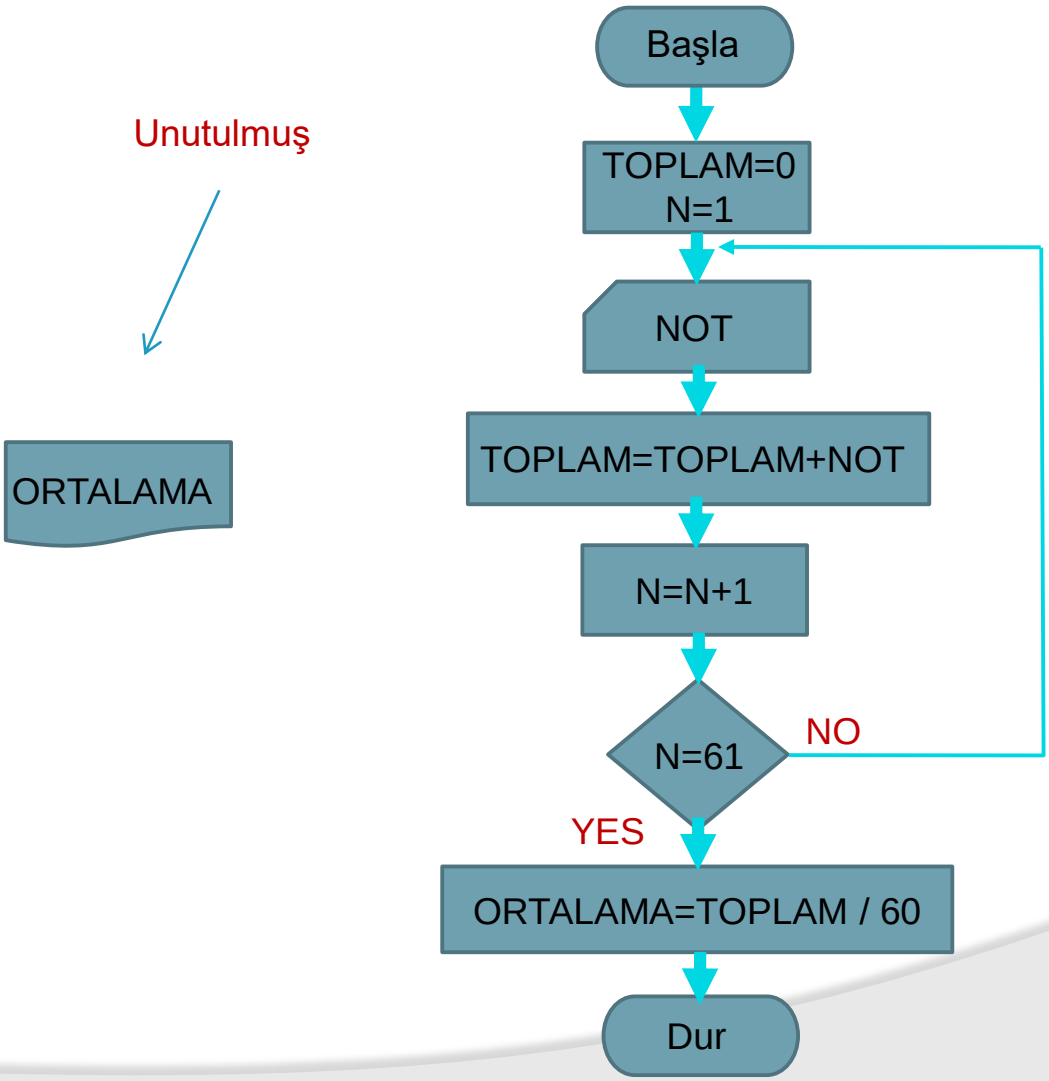
Bu örnek için acaba Akış Diyagramında yapılabilecek olası hatalar nelerdir?


Başla
TOPLAM=0
N=1
NOT
TOPLAM=TOPLAM+NOT
N=N+1
N=61
NO
YES
ORTALAMA=TOPLAM / 60
ORTALAMA
Dur


Başla
TOPLAM=0
N=0
NOT
TOPLAM=TOPLAM+NOT
N=N+1
N=61
NO
YES
ORTALAMA=TOPLAM / 60
ORTALAMA
Dur


Başla
TOPLAM=0
N=1
NOT
TOPLAM=TOPLAM+NOT
N=N+1
N=61
YES
NO
ORTALAMA=TOPLAM / 60
ORTALAMA
Dur


Başla
TOPLAM=0
N=1
NOT
TOPLAM=TOPLAM+NOT
N=N+1
N=61
NO
YES
ORTALAMA=TOPLAM / 60
ORTALAMA
Dur


Unutulmuş
ORTALAMA
Başla
TOPLAM=0
N=1
NOT
TOPLAM=TOPLAM+NOT
N=N+1
N=61
NO
YES
ORTALAMA=TOPLAM / 60
Dur

KAYNAKÇA

- Prof.Dr. İbrahim DEVELİ, Bilgisayar Programlama Ders Notları, Erciyes Üniv. Elektrik-Elektronik Müh. Böl.
- H.Turgut UYAR, Programlamaya Giriş Ders Notları,İTÜ, 2004.
- Fedon KADİFELİ,Standart C Programlama Dili, (Tercüme),2000.
- Doç. Dr. Soner ÇELİKKOL, Programlamaya Giriş ve Algoritmalar, Murathan Yayınevi, TRABZON; 2009
- N. Ercil Çağıltay ve ark., C DERSİ PROGRAMLAMAYA GİRİŞ, Ada Matbaacılık, ANKARA; 2009.
- Çeşitli kişilerin internette paylaşıma açtığı notlardan faydalanılmıştır.