

BilgisayarProgramlama (/github/mkarakoc/BilgisayarProgramlama/tree/master)  
/ dersnotlari (/github/mkarakoc/BilgisayarProgramlama/tree/master/dersnotlari)

## Hafıza ölçüm birimleri

<https://wiki.ubuntu.com/UnitsPolicy> (<https://wiki.ubuntu.com/UnitsPolicy>),

Aşağıdaki tanımlar IEC ve SI standartlarına göreler.

### bit (bits):

en küçük hafıza birimidir. Tek bir bilgi saklayabilir.

Saklaya bildiği bilgilerden biri genellikle "0" diğeri "1" ile temsil edilir.

### bayt (bytes):

1 byte = 8 bits

1 bayt = 8 bit

byte: bayt, [01.02.09], 04.05.08 (01.02.09)], Bir birim olarak ele alınan ve genellikle bir karakteri veya karakterin bir

parçasını temsil eden bitlerin sayısını içeren dizi. NOT 1: Belirli veri işleme sistemi için bir bayttaki bitlerin sayısı

sabittir. NOT 2: Genellikle bir bayt içindeki bit sayısı 8'dir.

### Kaynakça:

Bilişim Terimleri Sözlüğü (İngilizce - Türkçe)

<https://statik.tse.org.tr/upload/tr/dosya/icerikyonetimi/552/19012016170409-4.pdf>

(<https://statik.tse.org.tr/upload/tr/dosya/icerikyonetimi/552/19012016170409-4.pdf>)

Applications must use IEC standard for base-2 units:

- 1 KiB = 1.024 bytes (Note: big k)
- 1 MiB = 1.024 KiB = 1.048.576 bytes
- 1 GiB = 1.024 MiB = 1.048.576 KiB = 1.073.741.824 bytes
- 1 TiB = 1.024 GiB = 1.048.576 MiB = 1.073.741.824 KiB = 1.099.511.627.776 bytes

Applications must use SI standard for base-10 units:

- 1 kB = 1,000 bytes (Note: small k)
- 1 MB = 1,000 kB = 1,000,000 bytes
- 1 GB = 1,000 MB = 1,000,000 kB = 1,000,000,000 bytes
- 1 TB = 1,000 GB = 1,000,000 MB = 1,000,000,000 kB = 1,000,000,000,000 bytes

Prefixes  
for  
multiples  
of  
bits  
(/wiki/Bit)  
(bit)  
or  
bytes  
(/wiki/Byte)  
(B)

Value	Decimal ( <u>/wiki/Decimal_prefix</u> )	IEC ( <u>/wiki/IEC_80000- SI_prefix</u> )	JEDEC memory size ( <u>/wiki/JEDEC_memory_size</u> )
1024	$2^{10}$	Kk ( <u>/wiki/Kibi-</u> )	—
1024 <sup>2</sup>	$2^{20}$	Mi ( <u>/wiki/Mebi-</u> )	—
1024 <sup>3</sup>	$2^{30}$	Gi ( <u>/wiki/Gibi-</u> )	—
1024 <sup>4</sup>	$2^{40}$	Ti ( <u>/wiki/Tebi-</u> )	—
1024 <sup>5</sup>	$2^{50}$	Pi ( <u>/wiki/Peta-</u> )	—
1024 <sup>6</sup>	$2^{60}$	Ei ( <u>/wiki/Exa-</u> )	—
1024 <sup>7</sup>	$2^{70}$	Zi ( <u>/wiki/Zetta-</u> )	—
1024 <sup>8</sup>	$2^{80}$	Yi ( <u>/wiki/Yota-</u> )	—

<https://physics.nist.gov/cuu/Units/binary.html> (<https://physics.nist.gov/cuu/Units/binary.html>)

#### Examples and comparisons with SI prefixes

one kibibit	1 Kibit = $2^{10}$ bit = <b>1024 bit</b>
one kilobit	1 kbit = $10^3$ bit = <b>1000 bit</b>
one mebibyte	1 MiB = $2^{20}$ B = <b>1 048 576 B</b>
one megabyte	1 MB = $10^6$ B = <b>1 000 000 B</b>
one gibibyte	1 GiB = $2^{30}$ B = <b>1 073 741 824 B</b>
one gigabyte	1 GB = $10^9$ B = <b>1 000 000 000 B</b>

## Hafıza büyüklüklerine göre saklanabilecek bilgi çeşitlilik miktarı

- 1 bit'lik hafızada iki farklı bilgi saklanabilir.  $2^1 = 2$
- 2 bit'lik hafızada ise 4 farklı bilgi saklanabilir.  $2^2 = 4$
- 1 bayt'lık yani 8 bit'lik hafızada 256 farklı bilgi saklanabilir.  $2^8 = 256$

- 1 KiB = 1,024 bayt = 8192 bit =  $2^{13}$  bit'lik hafızada  $2^{(2^{13})}$  farklı bilgi saklanabilir.

In [8]:

1024 \* 8

Out[8]:

8192

## ASCII Table

<http://www.asciitable.com/> (<http://www.asciitable.com/>)

<https://en.wikipedia.org/wiki/ASCII> (<https://en.wikipedia.org/wiki/ASCII>)

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

Source: [www.LookupTables.com](http://www.LookupTables.com)

128	Ç	144	É	160	á	176	ð	192	Ł	208	ł	224	α	240	≡
129	Û	145	æ	161	í	177	é	193	Ł	209	ł	225	β	241	±
130	é	146	Æ	162	ó	178	ë	194	Ŧ	210	ŧ	226	Γ	242	≥
131	â	147	ô	163	ú	179	ı	195	Ł	211	ł	227	π	243	≤
132	ä	148	ö	164	ñ	180	ı	196	—	212	ı	228	Σ	244	∫
133	à	149	ò	165	Ñ	181	ı	197	Ł	213	ł	229	σ	245	∫
134	â	150	û	166	ı	182	ı	198	Ł	214	ł	230	μ	246	+
135	ç	151	ù	167	°	183	ı	199	Ł	215	ł	231	τ	247	≈
136	ê	152	ÿ	168	ı	184	ı	200	Ł	216	ł	232	Φ	248	°
137	ë	153	Ö	169	ı	185	ı	201	ı	217	ı	233	Θ	249	.
138	è	154	Û	170	ı	186	ı	202	Ł	218	ı	234	Ω	250	.
139	ı	155	ı	171	½	187	ı	203	ı	219	ı	235	δ	251	√
140	ı	156	ı	172	¾	188	ı	204	ı	220	ı	236	∞	252	∞
141	ı	157	ı	173	ı	189	ı	205	ı	221	ı	237	φ	253	z
142	Ä	158	ı	174	«	190	ı	206	ı	222	ı	238	ε	254	ı
143	Å	159	f	175	»	191	ı	207	ı	223	ı	239	ı	255	ı

Source: [www.LookupTables.com](http://www.LookupTables.com)

In [13]:

```
# 2 tabanından 10 tabanına
#      12345678
print(int('00000000', 2))

#      12345678
print(int('11111111', 2))

#      12345678
print(int('11111110', 2))

#      1234567812345678
print(int('1111111111111111', 2))

# 10 tabanından 2 tabanına
print(bin(255))
```

```
0
255
254
65535
0b11111111
```

## Türkiye'nin İlk Standardı

<https://www.tse.org.tr/Hakkimizda?ID=547&ParentID=80> (<https://www.tse.org.tr/Hakkimizda?ID=547&ParentID=80>)

**Kiraz:** Tazesinin yüz elli dirhemi bir akçeye, üç günden sonra iki yüz dirhemi bir akçeye, daha sonra iki yüz elli dirhemi bir akçeye ve her üç günden sonra yüz dirhem artırılarak en son iki okkası bir akçeye satılacak.



## Metre Convention

[https://en.wikipedia.org/wiki/Metre\\_Convention](https://en.wikipedia.org/wiki/Metre_Convention) ([https://en.wikipedia.org/wiki/Metre\\_Convention](https://en.wikipedia.org/wiki/Metre_Convention))

Turkey 1875 Joined originally as the Ottoman Empire United States 1878

The Metre Convention (French: Convention du Mètre), also known as the Treaty of the Metre,[1] is an international treaty that was signed in Paris on 20 May 1875 by representatives of 17 nations (Argentina, Austria-Hungary, Belgium, Brazil, Denmark, France, Germany, Italy, Peru, Portugal, Russia, Spain, Sweden and Norway, Switzerland, Ottoman Empire, United States of America, and Venezuela). The treaty created the International Bureau of Weights and Measures (BIPM), an intergovernmental organization under the authority of the General Conference on Weights and Measures (CGPM) and the supervision of the International Committee for Weights and Measures (CIPM), that coordinates international metrology and the development of the metric system.

## Algoritma ve Akış Diyagramı

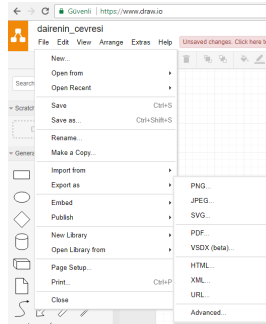
[https://en.wikipedia.org/wiki/Muhammad\\_ibn\\_Musa\\_al-Khwarizmi](https://en.wikipedia.org/wiki/Muhammad_ibn_Musa_al-Khwarizmi)  
([https://en.wikipedia.org/wiki/Muhammad\\_ibn\\_Musa\\_al-Khwarizmi](https://en.wikipedia.org/wiki/Muhammad_ibn_Musa_al-Khwarizmi))

**Algoritma:** Belirli bir problemi çözmek veya belirli bir amaca ulaşmak için aşamalarıyla beraber tasarlanan gerçekleştirme planı.

**Akış diyagramı:** Algoritmaların şematik hale getirilmesidir.

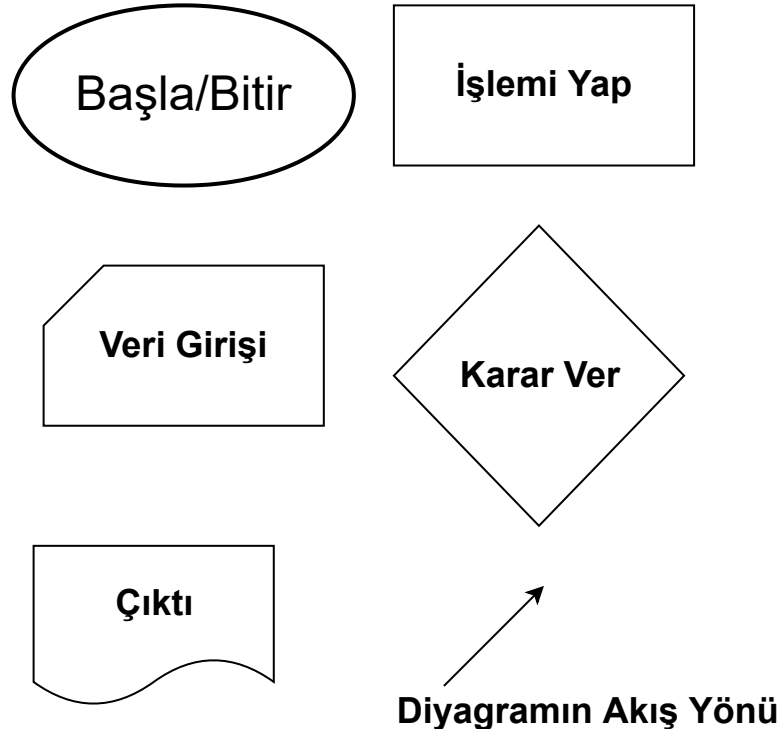
### Akış diyagramı çizimi

Aşağıdaki akış diyagramları "draw.io" diyagram (şema) çizim programı kullanılarak çizilmiştir. Çizilen şemalar aşağıda gösterildiği gibi "SVG" dosyası olarak kaydedilebilir.



Bağlantısı: <https://www.draw.io> (<https://www.draw.io>)

### Akış diyagramı kutucuklarının anlamları



### Bazı ilginç akış diyagramı siteleri

<https://www.101computing.net/from-flowcharts-to-python-code/> (<https://www.101computing.net/from-flowcharts-to-python-code/>)

<https://www.codeavengers.com/notes/planning/flowcharts>  
(<https://www.codeavengers.com/notes/planning/flowcharts>)

<https://www.pygame.org/project/355/3919> (<https://www.pygame.org/project/355/3919>)

<http://www.flowgorithm.org/about/index.htm> (<http://www.flowgorithm.org/about/index.htm>)

### Bilgisayar Programlama dersi ücretsiz (beleş)

Aşağıda aldığınız Bilgisayar programla dersinin ilk aşaması olan bir dersin web adresi verilmiştir. Devamında ise derse kayıt ücretini (Güncel kurdan 49 Dolar = 273.91 TL.) görmektesiniz. Dersinizin kıymetini bilmeniz ümidiyle...

#### Computational Thinking for Problem Solving (Problem Çözmede Hesaplı (Aşamalı) Düşünme)

<https://www.coursera.org/learn/computational-thinking-problem-solving>  
(<https://www.coursera.org/learn/computational-thinking-problem-solving>)

Computational thinking is the process of approaching a problem in a systematic manner and creating and expressing a solution such that it can be carried out by a computer. But you don't need to be a computer scientist to think like a computer scientist! ...

**Coursera**

Coursera is an online education company that partners with 160 of the world's top universities and industry leaders to offer courses, Specializations, and degrees that empower learners around the world to achieve their career, educational, and personal enrichment goals throughout their lives...

## Computational Thinking for Problem Solving

☒ Purchase Course · \$49 USD



Commit to earning a Certificate—it's a trusted, shareable way to showcase your new skills.

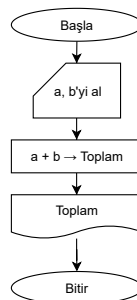
Continue

### Algoritma örnekleri:

**Örnek 1:** Bir bilgisayar kullanıcısı tarafından verilen, iki sayının toplamını bulmayı sağlayan algoritmayı yazınız.

1. Başla
2. Kullanıcıdan "a" ve "b" sayılarını al.
3. "a" ve "b"yi topla "Toplam"a aktar.
4. "Toplam"ı yazdır.
5. Bitir

Yukarıdaki örneğin akış diyagramı:





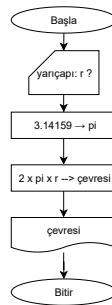
In [21]:

```
# *****  
# PYTHON 3 kullanılacak  
# *****  
  
# yukarıdaki algoritmayı uygulayan Python programını yaz.  
a = float(input('a = ?'))  
b = float(input('b = ?'))  
  
Toplam = a + b  
  
print(Toplam)
```

a = ? 4b = ? 59.0

**Örnek 2:** Bir dairenin çevresini hesaplamak için gerekli algoritmayı tasarlayınız.

1. Başla
2. Dairenin yarı çapı (r) nedir?
3. pi sayısını tanımla (3.14159-->pi).
4.  $2 \times \pi \times r$  --> çevresi
5. çevresini yazdır.
6. Bitir.



In [6]:

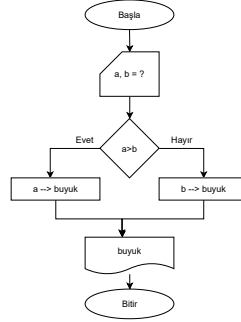
```
r = float(input('r=?'))  
pi = 3.14159  
cevre = 2*pi*r  
print(cevre)
```

r=? 212.56636

**Örnek 3:** a ve b gibi birbirinden farklı iki sayıdan büyük olanın yazılmasını sağlayan algoritmayı yazınız.

1. Basla.

2. a ve b sayılarını kullanıcıdan al.
3. Eğer  $a > b$  ise a --> büyük değilse b --> büyük
4. büyük'ün değerini yazdır.
5. Bitir.



In [4]:

```
a = float(raw_input('a = ?'))
b = float(raw_input('b = ?'))

if a > b:
    büyük = a
else:
    büyük = b

print büyük
```

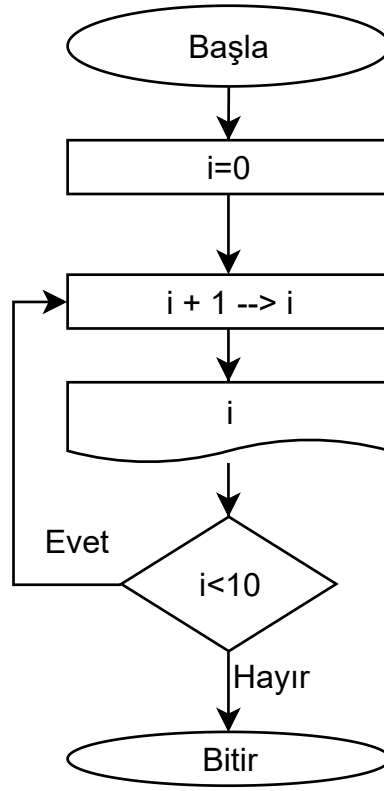
a = ? 4 b = ? 4.54.5

---

**Örnek 4-a:** 1'den 10'a kadar saymak için gerekli algoritmayı yazınız.

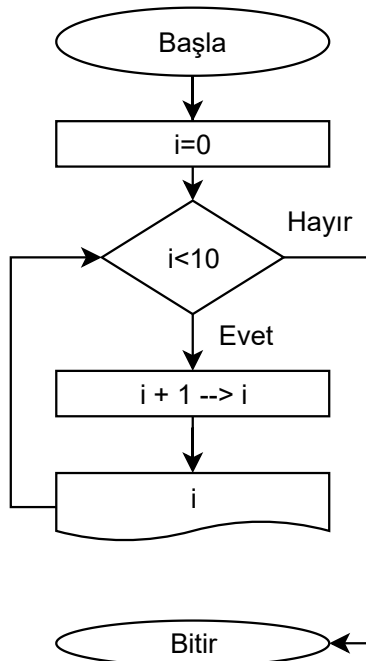
---

1. Başla
2. sayaç değişkeni i'nin ilk değer 0 olsun. ( $i = 0$ )
3.  $i + 1 \rightarrow i$
4. i'yi yazdır.
5. Eğer  $i < 10$  ise 3'e git değilse 6'ya devam et
6. Bitir.



**Örnek 4-b:** 1'den 10'a kadar saymak için gerekli algoritmayı yazınız.

1. Başla
2. sayaç değişkeni i'nin ilk değeri 0 olsun. ( $i = 0$ )
3. Eğer  $i < 10$  ise 4'e devam et değilse 7'ye git
4.  $i + 1 \rightarrow i$  ( $i = i + 1$ )
5. i'yi yazdır.
6. 3'e git
7. Bitir.



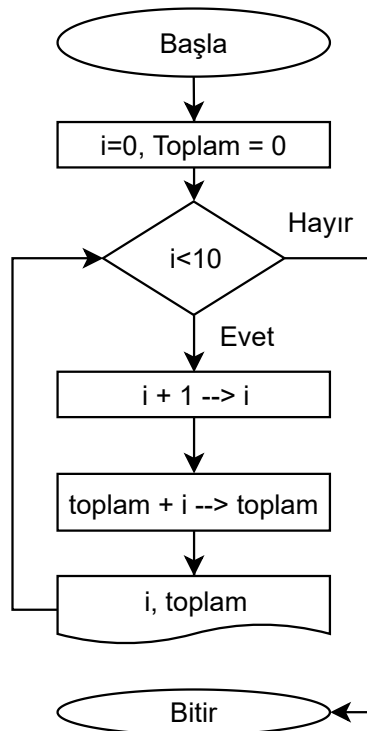
In [12]:

```
i=0
while i<10:
    i = i + 1
    print(i)
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

**Örnek 5:** 1'den 10'a kadar olan tam sayıları toplamak için gerekli algoritmayı yazınız.

1. Başla
2. sayaç değişkeni i'nin ilk değer 0 olsun. ( $i = 0$ )
3. toplam = 0
4. Eğer  $i < 10$  ise 4'e devam et değilse 8'e git
5.  $i + 1 \rightarrow i$  ( $i = i + 1$ )
6. toplam + i  $\rightarrow$  toplam
7. i'yi ve toplam'ı yazdır.
8. 3'e git
9. Bitir.



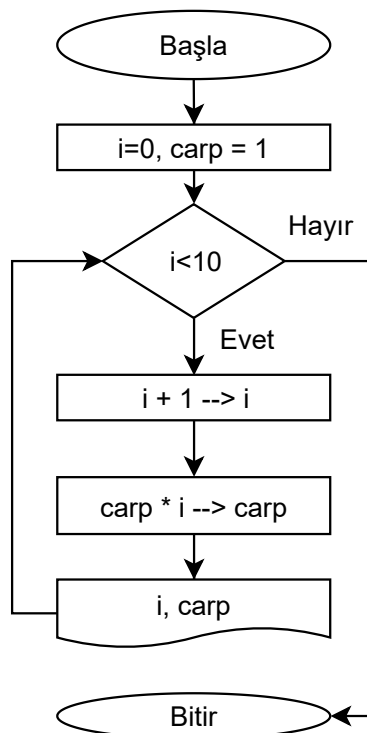
In [6]:

```
i = 0
toplam = 0
while i<10:
    i = i + 1
    toplam = toplam + i
    print i, toplam
```

```
1 1
2 3
3 6
4 10
5 15
6 21
7 28
8 36
9 45
10 55
```

**Örnek 6:** 1'den 10'a kadar olan tam sayıları birbiriyle çarpmak için gerekli algoritmayı yaz.

1. Başla
2. sayaç değişkeni i'nin ilk değeri 0 olsun. ( $i = 0$ )
3.  $carp = 1$
4. Eğer  $i < 10$  ise 4'e devam et değilse 8'e git
5.  $i + 1 \rightarrow i$  ( $i = i + 1$ )
6.  $carp * i \rightarrow carp$
7. i'yi ve carp'ı yazdır.
8. 3'e git
9. Bitir.



In [8]:

```
i = 0
carp = 1
while i<10:
    i = i + 1
    carp = carp * i
    print i, carp
```

```
1 1
2 2
3 6
4 24
5 120
6 720
7 5040
8 40320
9 362880
10 3628800
```

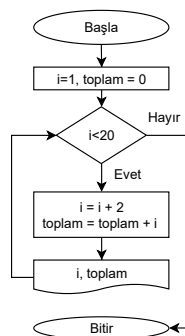
**Örnek 7-a:** [0 , 20] aralığındaki bütün tek tamsayıların toplamını elde edebilen algoritmayı geliştiriniz ve bu algoritmanın akış diyagramını çiziniz.

In [7]:

```
print "aradığımız cevap bu ama algoritma bu değil!"
print 1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19
```

```
aradığımız cevap bu ama algoritma bu değil!
100
```

1. Başla
2. sayaç değişkeni i'nin ilk değeri 1 olsun. ( $i = 1$ )
3. toplam = 0
4. Eğer  $i < 20$  ise 5'e devam et değilse 9'e git
5. toplam = toplam + i
6.  $i = i + 2$
7. i'yi ve toplam'ı yazdır.
8. 3'e git
9. Bitir.



In [45]:

```
i = 1
toplam = 0
while i<20:
    toplam = toplam + i
    print (i, toplam)
    i = i + 2
```

```
1 1
3 4
5 9
7 16
9 25
11 36
13 49
15 64
17 81
19 100
```

---

**Örnek 7-b:**  $a, b$  birer çift sayı ve  $a < b$  olmak üzere,  $[a, b]$  aralığındaki bütün tek tamsayıların toplamını döngü kullanmadan elde edebilecek bir algoritma geliştiriniz.

---

In [31]:

```
print("Bütün Sayılar")
for i in range(21):
    print(i, end=", ")

print("\n\nTek Sayılar", end="\n")
for i in range(1, 21,2):
    print(i, end=", ")

print("\n\nÇift Sayılar", end="\n")
for i in range(0, 21,2):
    print(i, end=", ")
```

Bütün Sayılar

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,

Tek Sayılar

1, 3, 5, 7, 9, 11, 13, 15, 17, 19,

Çift Sayılar

0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

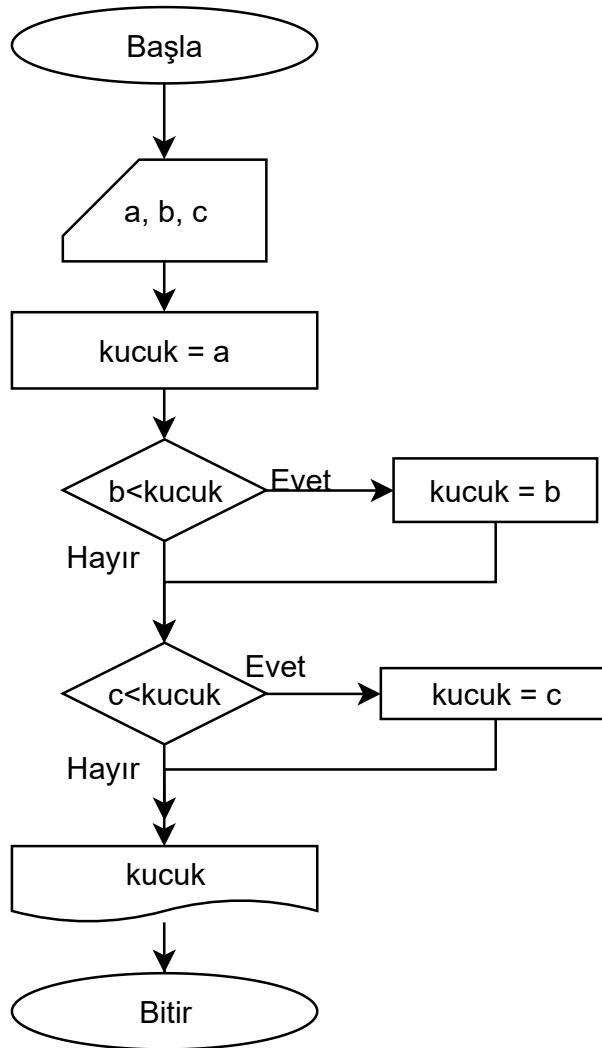
In [44]:

```
m = 10000
toplam = m**2/4
print(toplam)
```

25000000.0

**Örnek 8:** a, b ve c gibi bir birinden farklı üç sayıyı kullanıcıdan alıp en küçüğünü bulan algoritmayı geliştiriniz ve bu algoritmanın akış diyagramını çiziniz.

1. Başla
2. a, b, c sayılarını al.
3. a --> kucuk olsun.
4. Eğer  $b < \text{kucuk}$  ise  $\text{kucuk} = b$  (değilse 5'e git.)
5. Eğer  $c < \text{kucuk}$  ise  $\text{kucuk} = c$  (değilse 6'ya git.)
6. kucuk'ün değerini yazdır.
7. Bitir





In [105]:

```
a = float(input('a = ?'))
b = float(input('b = ?'))
c = float(input('c = ?'))

kucuk = a
if b < kucuk: kucuk = b

if c < kucuk: kucuk = c

print ("Üç sayının en küçüğü %s dir."%kucuk)
```

a = ? 3b = ? 1c = ? 2Üç sayının en küçüğü 1.0 dir.

---

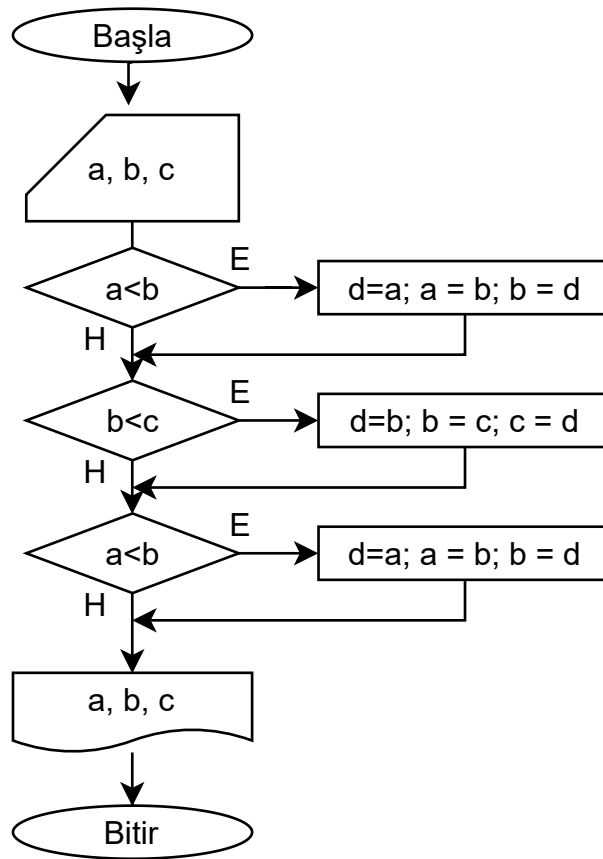
**Örnek 9:** a, b, c gibi üç farklı sayıyı büyükten küçüğe doğru sıralayan bir algoritma geliştiriniz.

---

Eğer a, b, c değerleri 1, 2, 3 sayılarından herhangi biriye olasılıklar tablosu aşağıdaki gibidir.

a	b	c
1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1

1. Başla.
2. a, b, c sayılarını al.
3. Eğer  $a < b$  ise degistir = a; a = b; b = degistir
4. Eğer  $b < c$  ise degistir = b; b = c; c = degistir
5. Eğer  $a < b$  ise degistir = a; a = b; b = degistir
6. a, b, c 'yi yaz.
7. Bitir



In [106]:

```

a = float(input('a = ?'))
b = float(input('b = ?'))
c = float(input('c = ?'))

if a<b: degistir = a; a = b; b = degistir
if b<c: degistir = b; b = c; c = degistir
if a<b: degistir = a; a = b; b = degistir

print(a, b, c)

```

a = ? 1b = ? 2c = ? 33.0 2.0 1.0

```

if a<b: degistir = a; a = b; b = degistir

```

yerine

```

if a<b: b, a = a, b

```

kullanılabilir.

In [102]:

```
a = float(input('a = ?'))
b = float(input('b = ?'))
c = float(input('c = ?'))

if a<b: b, a = a, b
if b<c: c, b = b, c
if a<b: b, a = a, b

print(a, b, c)
```

a = ? 4b = ? 12c = ? 512.0 5.0 4.0

In [21]:

```
# dört sayıyı büyükten küçüğe sıralayan Python programı
a = float(raw_input('a = ?'))
b = float(raw_input('b = ?'))
c = float(raw_input('c = ?'))
d = float(raw_input('d = ?'))

i = 0
while i<3:
    i = i + 1
    print i
    if a<b: b, a = a, b
    if b<c: c, b = b, c
    if c<d: d, c = c, d

print(a, b, c, d)
```

a = ? 1b = ? 2c = ? 3d = ? 41  
2  
3  
4.0 3.0 2.0 1.0

---

**Örnek 9:** 5 elemanlı bir listedeki tam sayıları küçükten büyüğe doğru sıralayan bir algoritma geliştiriniz.

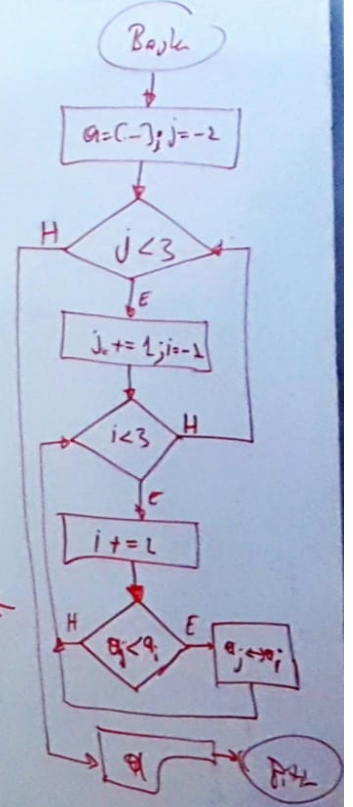
---

**Çözüm:**

- Rastgele 5 tane sayıdan oluşan bir listeyi küçükle büyüğe sıralayan algoritmayı geliştiriniz.
- $a = [9, 3, 5, -2]$   $\rightarrow a' = [-2, 3, 5, 9]$   
 $a = [a_0, a_1, a_2, a_3]$   $\rightarrow a_0 = 9, \dots, a_3 = -2$

1. Boyları
2.  $a = [\dots]$ ,  $j = -1$
3.  $j < 3$  ise devam değilde 8. g. t.
4.  $j += 1$ ;  $i = -1$
5.  $i < 3$  ise devam değilde 3. g. t.
6.  $i += 1$
7.  $a_j > a_i$  ise  $dg = a_j$ ;  $a_j = a_i$ ;  $a_i = dg$  değilde 5. g. t.
8.  $a$ 'yı göster
9. B.t.r.

i	i < 3	i++
-1	T	0
0	T	1
1	T	2
2	T	3
3	F	...



## Python programı

In [1]:

```

a = [9, 3, 5, -2, 0]
print(a)

j = -1
while j < 4:
    j += 1
    i = -1
    while i < 4:
        i += 1
        if a[j] < a[i]:
            dg = a[j]
            a[j] = a[i]
            a[i] = dg

print(a)

```

```

[9, 3, 5, -2, 0]
[-2, 0, 3, 5, 9]

```

Algoritmanın doğruluğunu test eden tablo (doğruluk tablosu)

In [4]:

```
a = [9, 3, 5, -2, 0]
j = -1
sira = 0

print ("{: ^10}{: >3}{: ^7}{: ^2}{: >2}{: ^7}{: >3}{: ^12}{: ^16}".format("Sıra No", "j", "j<3"
while j<4:
    if j<4:
        jb = "Doğru"
    else:
        jb = "Yanlış"
    j += 1
    i=-1
    while i<4:
        if i<4:
            ib = "Doğru"
        else:
            ib = "Yanlış"
        i+=1
        ab = a[j]<a[i]

        if ab:
            abt = "Doğru"
        else:
            abt = "Yanlış"

        sira += 1
        print ("      {:02d}      {:3d} {}{:3d}{:3d} {}{:3d} {: ^10} {}".format(sira, j-1,
        if ab:
            a[j], a[i] = a[i], a[j]
print()
```

Sıra No	j	j<3	j+1	i	i<3	i+1	a[j]<a[i]	liste
01	-1	Doğru	0	-1	Doğru	0	Yanlış	[9, 3, 5, -2, 0]
02	-1	Doğru	0	0	Doğru	1	Yanlış	[9, 3, 5, -2, 0]
03	-1	Doğru	0	1	Doğru	2	Yanlış	[9, 3, 5, -2, 0]
04	-1	Doğru	0	2	Doğru	3	Yanlış	[9, 3, 5, -2, 0]
05	-1	Doğru	0	3	Doğru	4	Yanlış	[9, 3, 5, -2, 0]
06	0	Doğru	1	-1	Doğru	0	Doğru	[9, 3, 5, -2, 0]
07	0	Doğru	1	0	Doğru	1	Yanlış	[3, 9, 5, -2, 0]
08	0	Doğru	1	1	Doğru	2	Yanlış	[3, 9, 5, -2, 0]
09	0	Doğru	1	2	Doğru	3	Yanlış	[3, 9, 5, -2, 0]
10	0	Doğru	1	3	Doğru	4	Yanlış	[3, 9, 5, -2, 0]
11	1	Doğru	2	-1	Doğru	0	Yanlış	[3, 9, 5, -2, 0]
12	1	Doğru	2	0	Doğru	1	Doğru	[3, 9, 5, -2, 0]
13	1	Doğru	2	1	Doğru	2	Yanlış	[3, 5, 9, -2, 0]
14	1	Doğru	2	2	Doğru	3	Yanlış	[3, 5, 9, -2, 0]
15	1	Doğru	2	3	Doğru	4	Yanlış	[3, 5, 9, -2, 0]
16	2	Doğru	3	-1	Doğru	0	Doğru	[3, 5, 9, -2, 0]
17	2	Doğru	3	0	Doğru	1	Doğru	[-2, 5, 9, 3, 0]
18	2	Doğru	3	1	Doğru	2	Doğru	[-2, 3, 9, 5, 0]
19	2	Doğru	3	2	Doğru	3	Yanlış	[-2, 3, 5, 9, 0]
20	2	Doğru	3	3	Doğru	4	Yanlış	[-2, 3, 5, 9, 0]

21	3	Doğru	4	-1	Doğru	0	Yanlış	[-2, 3, 5, 9, 0]
22	3	Doğru	4	0	Doğru	1	Doğru	[-2, 3, 5, 9, 0]
23	3	Doğru	4	1	Doğru	2	Doğru	[-2, 0, 5, 9, 3]
24	3	Doğru	4	2	Doğru	3	Doğru	[-2, 0, 3, 9, 5]
25	3	Doğru	4	3	Doğru	4	Yanlış	[-2, 0, 3, 5, 9]

**Aynı algoritmanın biraz değiştirilmiş halinin doğruluk tablosu\***

In [34]:

```
a = [9, 3, 5, -2, 0]
Na = len(a) - 1
j = -1
sira = 0

print ("{: ^10}{: >3}{: ^7}{: ^2}{: >2}{: ^7}{: >3}{: ^12}{: ^16}".format("Sıra No", "j", "j<3"
while j<Na:
    if j<Na:
        jb = "Doğru"
    else:
        jb = "Yanlış"
    j += 1
    i=-1
    while i<Na-1:
        if i<Na-1:
            ib = "Doğru"
        else:
            ib = "Yanlış"
        i+=1
        ab = a[i]>a[i+1]

        if ab:
            abt = "Doğru"
        else:
            abt = "Yanlış"

        sira += 1
        print ("      {:02d}      {:3d} {}{:3d}{:3d} {}{:3d} {: ^10} {}".format(sira, j-1,
        if ab:
            a[i], a[i+1] = a[i+1], a[i]
print()
```

Sıra No	j	j<3	j+1	i	i<3	i+1	a[j]<a[i]	liste
01	-1	Doğru	0	-1	Doğru	0	Doğru	[9, 3, 5, -2, 0]
02	-1	Doğru	0	0	Doğru	1	Doğru	[3, 9, 5, -2, 0]
03	-1	Doğru	0	1	Doğru	2	Doğru	[3, 5, 9, -2, 0]
04	-1	Doğru	0	2	Doğru	3	Doğru	[3, 5, -2, 9, 0]
05	0	Doğru	1	-1	Doğru	0	Yanlış	[3, 5, -2, 0, 9]
06	0	Doğru	1	0	Doğru	1	Doğru	[3, 5, -2, 0, 9]
07	0	Doğru	1	1	Doğru	2	Doğru	[3, -2, 5, 0, 9]
08	0	Doğru	1	2	Doğru	3	Yanlış	[3, -2, 0, 5, 9]
09	1	Doğru	2	-1	Doğru	0	Doğru	[3, -2, 0, 5, 9]
10	1	Doğru	2	0	Doğru	1	Doğru	[-2, 3, 0, 5, 9]
11	1	Doğru	2	1	Doğru	2	Yanlış	[-2, 0, 3, 5, 9]
12	1	Doğru	2	2	Doğru	3	Yanlış	[-2, 0, 3, 5, 9]
13	2	Doğru	3	-1	Doğru	0	Yanlış	[-2, 0, 3, 5, 9]
14	2	Doğru	3	0	Doğru	1	Yanlış	[-2, 0, 3, 5, 9]
15	2	Doğru	3	1	Doğru	2	Yanlış	[-2, 0, 3, 5, 9]
16	2	Doğru	3	2	Doğru	3	Yanlış	[-2, 0, 3, 5, 9]
17	3	Doğru	4	-1	Doğru	0	Yanlış	[-2, 0, 3, 5, 9]
18	3	Doğru	4	0	Doğru	1	Yanlış	[-2, 0, 3, 5, 9]
19	3	Doğru	4	1	Doğru	2	Yanlış	[-2, 0, 3, 5, 9]

20      3 Doğru   4   2 Doğru   3   Yanlış   [-2, 0, 3, 5, 9]

**N** tane tam sayıyı sıralayabilen genelleştirilmiş Python programı.

In [41]:

```
import numpy as np

a = list(np.random.randint(-500, 500, 50))

Na = len(a) - 1
print("Rastgele {} tam sayı".format(Na+1))
print(a)

j = -1
while j<Na:
    j+=1
    i=-1
    while i<Na-1:
        i+=1
        if a[i]>a[i+1]:
            a[i], a[i+1] = a[i+1], a[i]

print("\nKüçükten büyüğe sıralanmış {} tam sayı".format(Na+1))
print(a)
```

Rastgele 50 tam sayı

[-309, -206, 482, 438, -271, 18, 258, -275, -317, -203, 392, 5, 261, 3

Küçükten büyüğe sıralanmış 50 tam sayı

[-476, -447, -387, -360, -317, -315, -309, -294, -275, -271, -266, -24

## Fibonacci Sayıları - Doğruluk tablosu



In [3]:

```
n = int(input("n=?"))

f0, f1 = 0, 1
i = 1
print ("-"*24)
print ("i  i<n  i+1  f0  f1  f2")
print ("-"*24)
while i<n:
    ib = i<n
    i += 1
    f2 = f1 + f0
    print ("{} {} {} {} {}".format(i-1, ib, i, f0, f1, f2))
    f0, f1 = f1, f2

ib = i<n
print ("{} {} {} - - -".format(i, ib, i+1))
```

```
n=? 6-----
i  i<n  i+1  f0  f1  f2
-----
1  True  2    0  1  1
2  True  3    1  1  2
3  True  4    1  2  3
4  True  5    2  3  5
5  True  6    3  5  8
6  False 7    -  -  -
```

In [0]: