



Visual Studio® ile

AZ VE ÖZ C++



Muhammet ÇAĞATAY

DÜZENLEME: 3

04.01.2014

BİLGİSAYAR VE PROGRAMLAMA DİLİ HAKKINDA GENEL BİLGİLER.....	3
C++ DERLEYİCİLERİ.....	4
VİSUAL STUDIO İLE YENİ BİR C++ PROJESİ OLUŞTURMAK.....	4
C++ İSKELET YAPISI VE TEMEL C++ KODLARI.....	7
KODLARIN DERLENİP ÇALIŞTIRILMASI.....	8
C++ DİLİNDE AÇIKLAMA SATIRLARI.....	8
COUT (ÇIKIŞ) KOMUTUNUN KULLANILMASI	8
DEĞİŞKEN TANIMI VE DEĞİŞKEN TANIMLAMA KURALLARI.....	10
DEĞİŞKEN TİPLERİ.....	11
CİN (GİRİŞ) KOMUTUNUN KULLANILMASI	12
BASİT MATEMATİKSEL OPERATÖRLER	13
MANTIKSAL OPERATÖRLERİ.....	13
ARTIŞ AZALIŞ OPERATÖRLERİ	13
HIZLI İŞLEM OPERATÖRLERİ	13
ALGORİTMA VE PROGRAMLAMAYA GİRİŞ	14
ALGORİTMA AKIŞ DİYAGRAMLARI.....	14
CMATH (MATEMATİK) KÜTÜPHANESİ.....	15
MATEMATİKSEL FONKSİYONLAR.....	15
PRINTF ÇIKIŞ KOMUTUNUN KULLANIMI... ..	16
İF KOŞUL YAPISI.....	18
İF – ELSE İF KOŞUL YAPISI.....	21
İF – ELSE İÇİN KISA KONTROL YAPISI.....	22
VERİ TİPLERİNİ DÖNÜŞTÜRMEK.....	22
WHİLE DÖNGÜSÜ.....	23
DO WHİLE DÖNGÜSÜ.....	23

GOTO YAPISI.....	23
FOR DÖNGÜSÜ.....	25
BREAK DEYİMİ.....	26
CONTINUE DEYİMİ.....	26
DÖNGÜLER İLE İLGİLİ ÖRNEKLER.....	27
SWITCH CASE KONTROL YAPISI.....	33
FONKSİYONLAR (ALT PROGRAMLAR).....	36

GİRİŞ

Ben de dahil olmak üzere biz Türkler uzun uzun yazıları okumayı sevmiyoruz. Bu yüzden serinin adı “az ve öz” , bu ve diğer materyallerimde de her şey özetin özeti olarak, bu işe yeni başlayan arkadaşlarımıza destek olsun diye anlatılacaktır.

BİLGİSAYAR NEDİR ?

Çoğu sıradan insan için bilgisayar muazzam bir aygıt olsa da biz bilişicilerin bilgisayara verdiği isim **salak kutu**’dur. Bilgisayarların aklı yoktur, sadece verilen komutları yerine getirirler. Ne istersek yaptırabileceğimiz sadık makinelerdir ve Hollywood filmlerinde olduğu gibi günün birinde evreni ele geçirme olasılıkları sıfırdır.

BİLGİSAYARLAR NASIL ÇALIŞIR

Bilgisayarın temel çalışma prensibi **giriş >> işlem >> çıkış** şeklindedir.

Giriş: Bilgisayarın üzerinde çalışacağı veri kullanıcı tarafından girilir.

İşlem: Bilgisayar girilen bilgiyi işler.

Çıkış: Bilgisayar insanların anlayacağı şekilde ekrana sonucu gösterir.

Yani bilgisayarlar aslında verdiğiniz girdileri işleme sokup size sonucu gösterirler.

PEKİ BU MİLLETİN AĞZINA DOLANAN 0 VE 1 LER NEYİN NESİ ?

Bilgisayarlar elektronik aygıtlar olduğu için işlemleri elektronların hareketleri ile gerçekleştirirler. Elektroniğin devreden geçme durumu **1 (doğru)** , devreden elektronun geçmeme durumu ise **0 (yanlış)** olarak ifade edilir. Bu kavram lise yıllarımızda, matematik dersinde gördüğümüz önermeden başkası değildir.

Elektronikte 0 ve 1 (**açık / kapalı**) kapılarından sorumlu devlet bakanı dediğimiz bir devre elemanı vardır ve bu elemanın adı transistördür. Onlarca transistörler bir araya gelerek mikroçipleri, binlercesi ise bilgisayarın beyni olan işlemcileri oluşturur.

Bir işlemcide binlerce transistör olduğu için, 0 yada 1 mi şeklinde elimizde binlerce kombinasyon vardır. Bu kombinasyonların her biri işlemci için bir anlam içermekte ve işlemci o ikilik sistemdeki sayılara göre iş yapmaktadır. Bu ikililer bitleri, baytları, kilobaytları, megabaytları, gigabaytları, terabaytları... oluşturur.

PROGRAMLAMA DİLLERİ NEDİR ?

Bilgisayarların bildiği dil 0 ve 1 lerdir. Bizim bu makine dilini bilmemiz çok zordur, bu yüzden gündelik hayata benzer kodlarla bilgisayara komut vermemizi sağlayan kod yapıları geliştirilmiştir ve bunlara programlama dili denmiştir. Yazdığımız kodlar bu dillerin derleyicileri sayesinde 0 ve 1 lere dönüştürülmekte ve bilgisayar ne yaptırmak istediğimizi anlayıp ona göre iş yapmaktadır.

C++ PROGRAMLAMA DİLİ NEDİR ?

Bilgisayarın icadı ile a, b, c, basic... gibi birçok programlama dili üretilmiştir, günümüzde de 250 den fazla programlama dili olduğu söylenmektedir ve her dilin kendine özgü bir işlevi vardır.

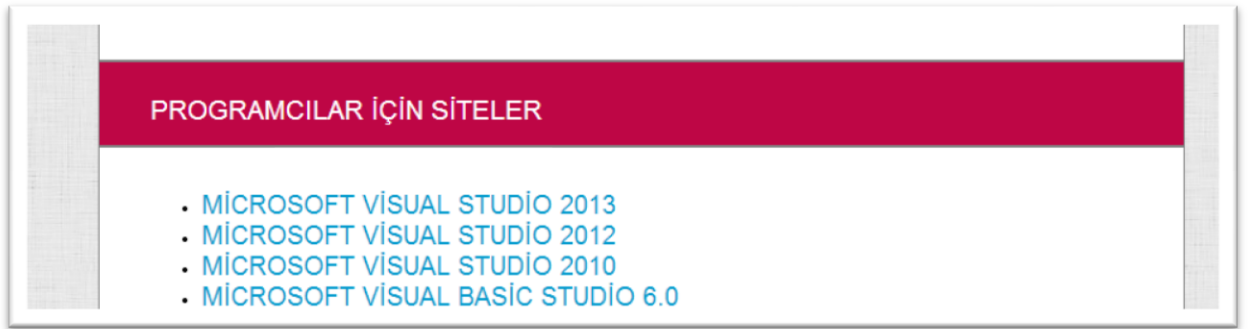
C++ dili ise; 1970 de üretilen c dilinin geliştirilmiş halidir. C dili ilk programlama dillerindendir ve oldukça gelişmiş işlevleri vardır.

C++, nesne tabanlı diye isimlendirdiğimiz bir dildir yani işler metod, sınıf , nesne diye isimlendirdiğimiz ve sizin de zamanla öğreneceğiniz yapılar üzerinden yürümektedir. C++ ne kadar gelişmiş ki diye soracak olursanız, çoğumuzun bilgisayarında yüklü olan **Microsoft Windows** işletim sistemi, c++ dili ile yazılmıştır, gerisini siz düşünün artık ☺

C++ DERLEYİCİLERİ

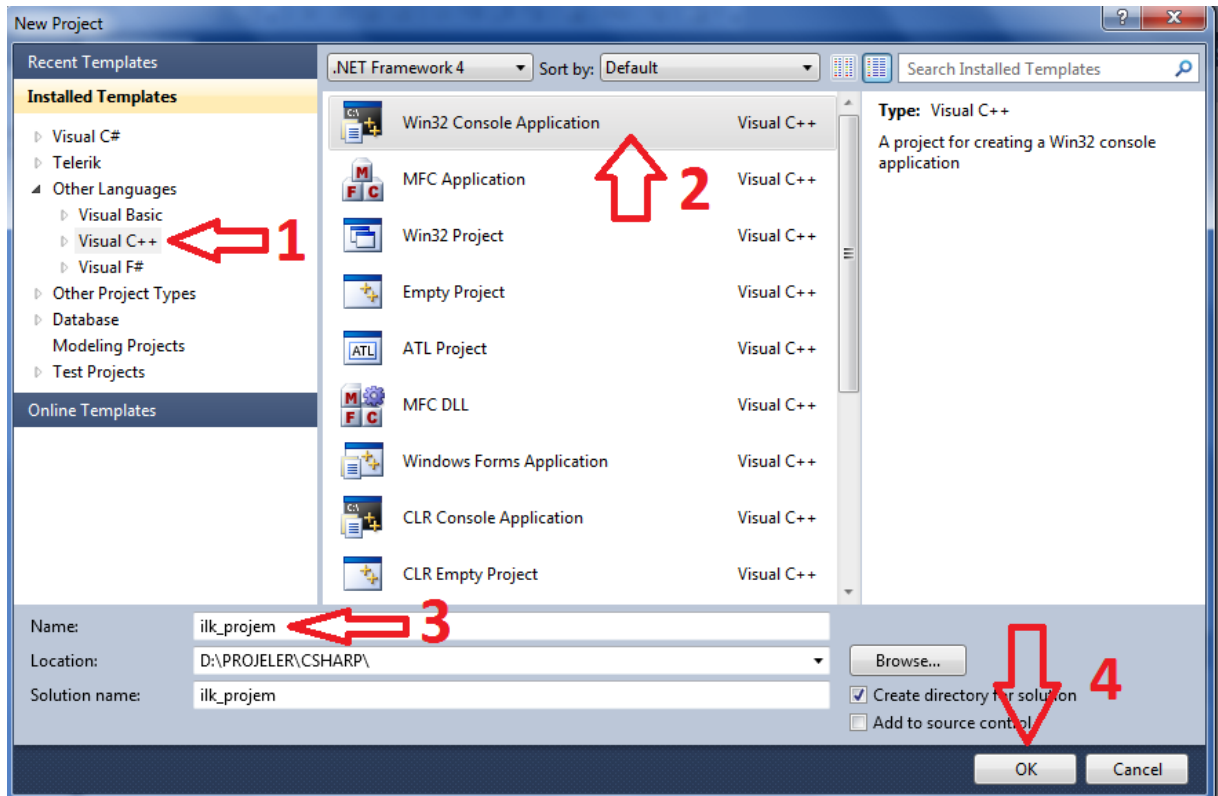
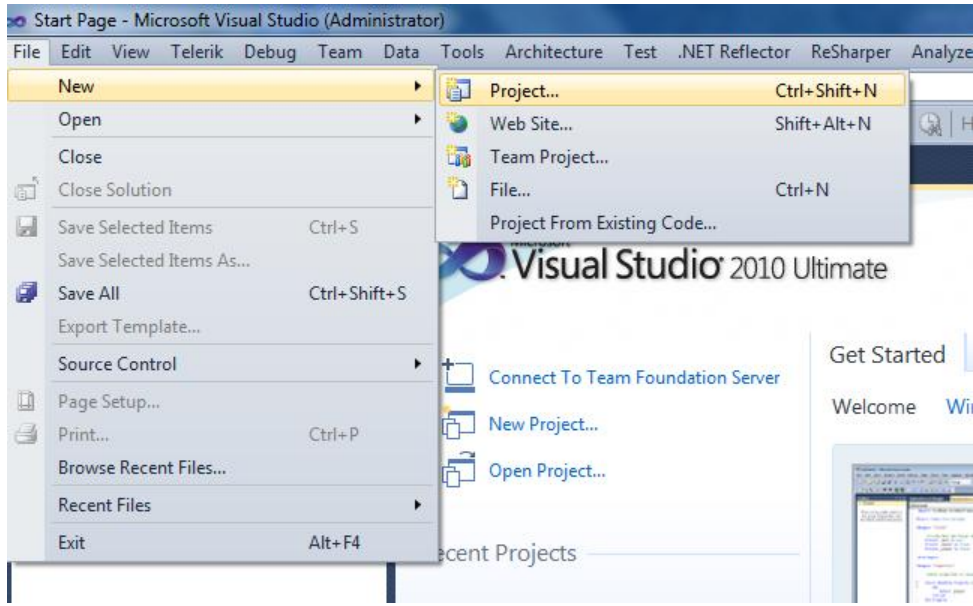
C++ ile program yazmak için çeşitli studiolar (programlar) vardır. Bunlardan en çok kullanılanlar **DEV C++** yada **MICROSOFT VISUAL STUDIO** dur. Dev c++ in kullanımı daha kolaydır ama ben size Microsoft visual studio 2010 yada üst versiyonlarını kullanmanızı öneririm ve bu eğitimde de ben visual studio 2010 sürümü üzerinden c++ dilini anlatmaya çalışacağım ☺

Gerekli programları <http://muhammetcagatay.com/sayfalar/programlar.html> adresinden edinebilirsiniz.

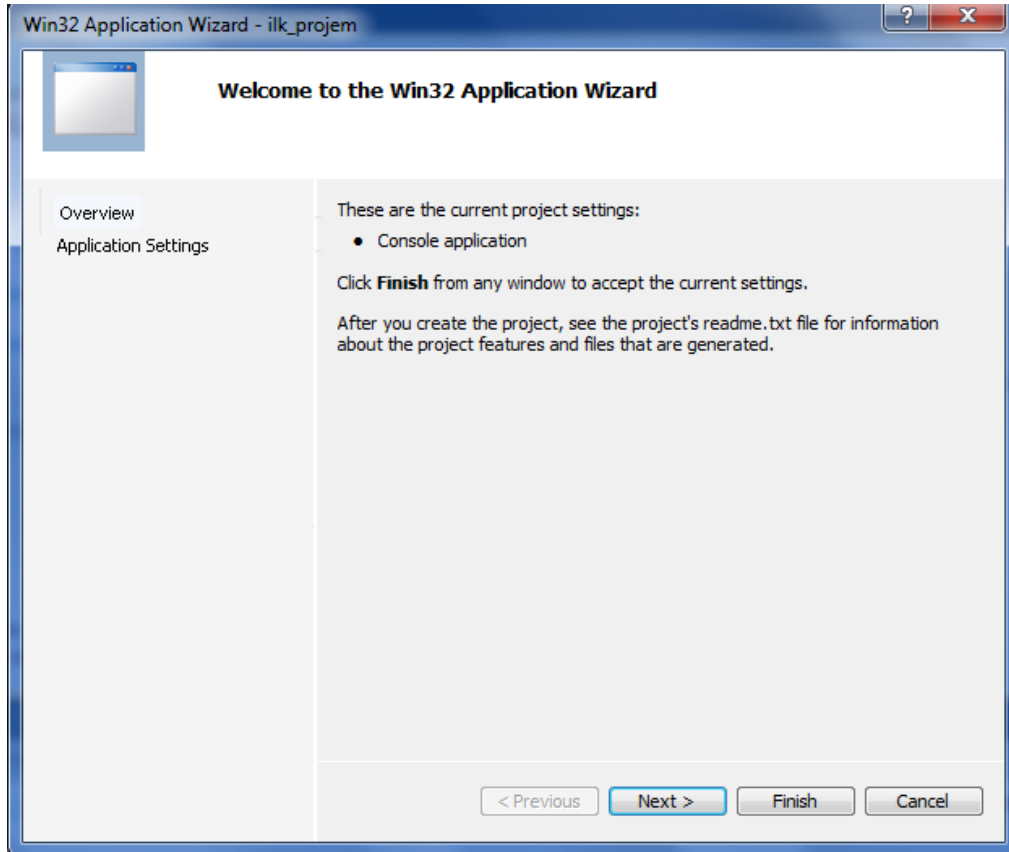


VISUAL STUDIO İLE YENİ BİR C++ PROJESİ OLUŞTURMA

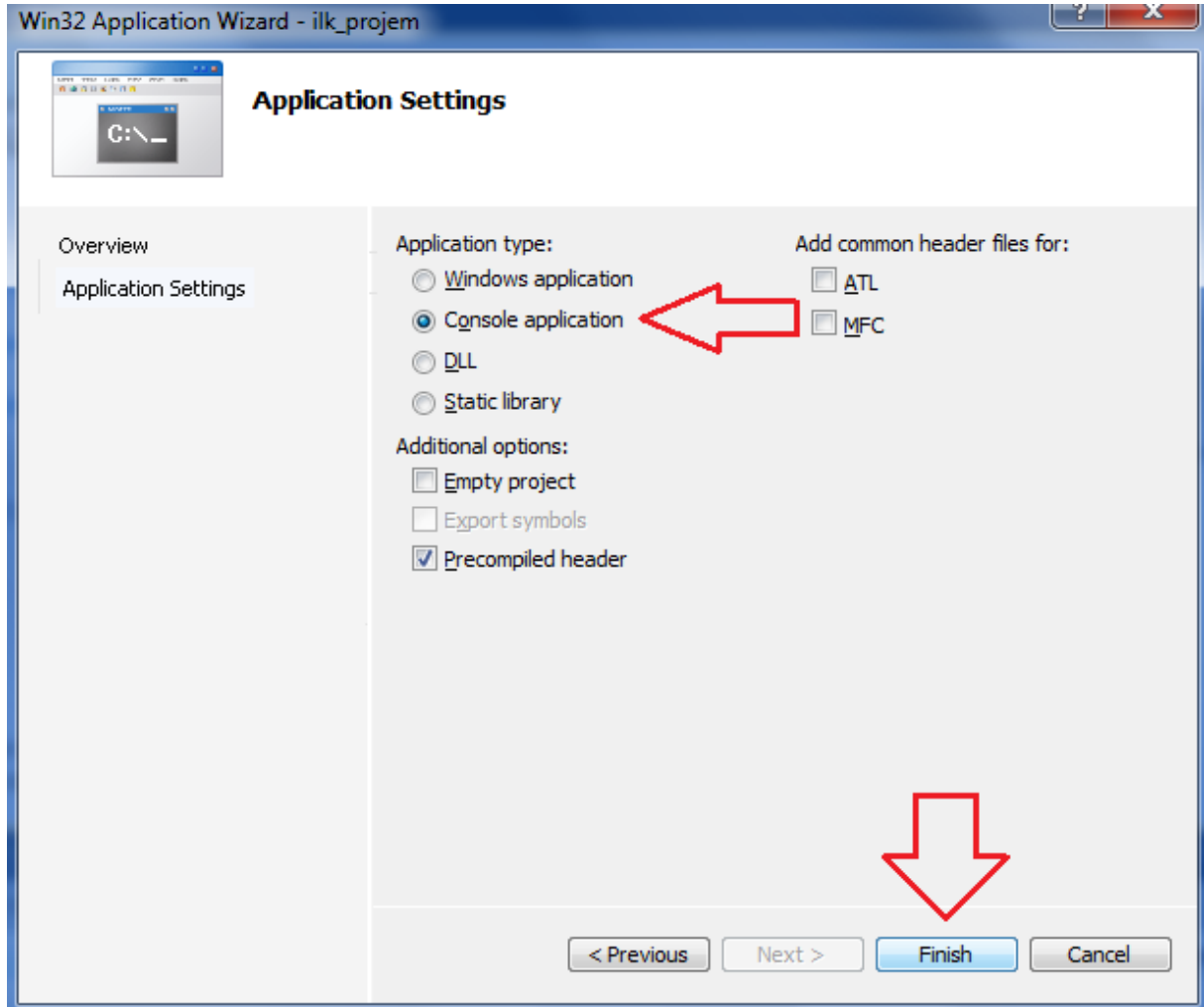
Stuidoyu açtıktan sonra **ctrl + shift + n** yada **file > new > project** ile yeni bir proje oluşturabilirsiniz.



Proje isimlerini verirken boşluk ve Türkçe karakter kullanmak yanlış değildir ama kullanmanızı ben tavsiye etmem.

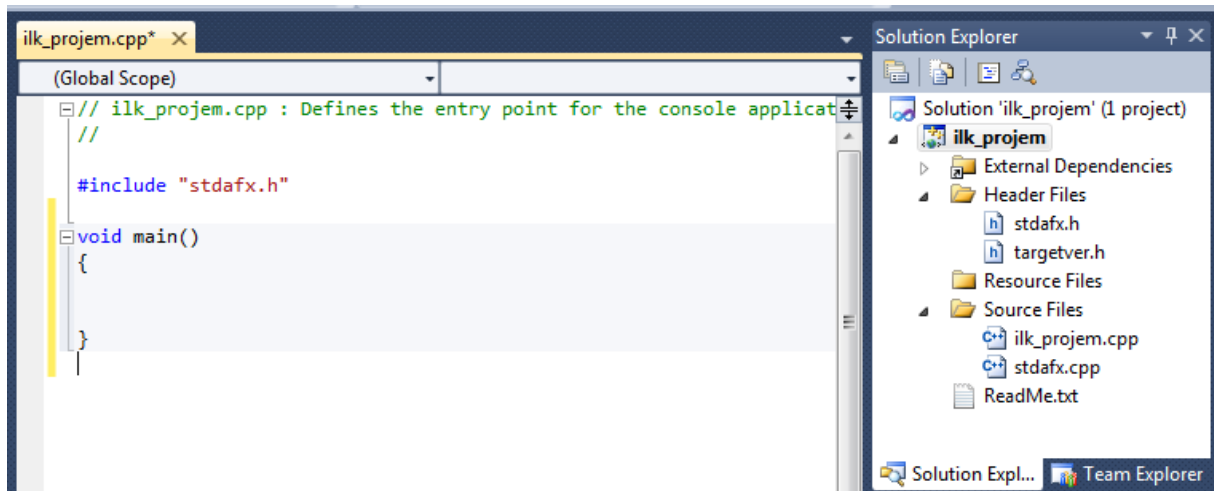


Adımları izledikten sonra gelen pencerede **next** tuşuna tıklayın



Ardından **finish** butonuna basarak projenizi oluşturun. Artık yazmaya başlayabiliriz.

c++ editörünün dosya uzantısı **.cpp** dir ve açılımı **c plus plus** tır.



C++ İSKELET YAPISI VE TEMEL C++ KODLARI


1. Programlama dili derleyicileri yazılan kodları ilk adımdan son adıma kadar sıra sıra okurlar.
2. Eğer kodlarda bir mantık ya da yazım hatası varsa, ekrana hata mesajını yazdırıp dururlar bu olaya *programın patlaması* diyoruz.
3. Eğer hata yoksa tek tek tüm komutları okuyup kodların sonucunda oluşan programı bize gösterirler.
4. C++ dilinde kodların sonuna “;” konur. Ve kodlarımız { } parantezleri (bloklar) arasına yazılır.

```
1 #include "stdafx.h"
2 #include <iostream>
3 using namespace std;
4
5 void main()
6 {
7     kodlar
8     system("pause");
9 }
```

C++ dilinin iskelet yapısı yandaki şekilde görüldüğü gibidir.

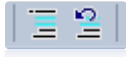
#include :	İngilizcede dâhil etmek anlamına gelmektedir. C++ dilinde ise derleyicinin içinde bulunan hazır kütüphaneleri projemize dâhil etmek için kullanılır.
<iostream> :	İ = input yani giriş (<i>dışarıdan programa değer girmek</i>) ve o = output yani çıkış (<i>sonuçların ekrana yazdırılması</i>) işlemlerinin tutulduğu temel c++ kütüphanesidir.
std :	<p>Standart kelimesinin kısaltılmış halidir. C++ ın tüm standart kütüphane bileşenlerinin tanımlı olduğu bir isim uzayıdır.</p> <p>Eğer using namespace std; isim uzayını uygulamamıza eklemesek;</p> <pre>#include <iostream> void main () { using std::cin; using std::cout; using std::endl; ... }</pre> <p>Yukarıdaki örnekte görüldüğü gibi, programımızda kullanacağımız bütün c++ kütüphanelerini, kodlarımıza teker teker eklemek zorunda kalırız.</p>
main() :	Programın ana metodudur. C++ kodları çalıştırıldığında main metodundan başlarlar, eğer main metodu yoksa program nerden başlayacağını bilemez ve durur.
system("pause") :	Programımızı derlediğimizde çalışır, sonuç ekrana gelir ve kapanır. Bu süreç çok hızlı olduğu için bizim gözümüz yetişemeyebilir ve programın çıktısını göremeyebiliriz. Bu yüzden kod parçalarımızın sonuna system("pause") yazarak geçici süreliğine (<i>biz bir tuşa basıncaya dek</i>) sistemi durdururuz ve sonuçları daha sağlıklı bir şekilde görürüz.
"stdafx.h"	Microsoft'un header dediğimiz gövde kütüphanesidir, bizimle bir alakası olmayan bu kod derleme esnasında studio tarafından kullanılır. Bunu yazmadığımız takdirde programımız çalışmayabilir.


KODLARIN DERLENİP ÇALIŞTIRILMASI

Editöre kodlarımız yazdıktan sonra derlemek için **F5** tuşunu ya da  ikonunu kullanabilirsiniz.

C++ DİLİNDE AÇIKLAMA SATIRLARI

1. Açıklama satırları kodlarımızın yanına hatırlatıcı notlar ya da kod ile ilgili bilgi vermek için kullanılan yazılardır.
2. C++ da açıklama yapmak için satırın başına “ **//** ” konulmalıdır.
3. C++ editöründe açıklama satırları **yeşil** renkte görünür ve C++ derleyicisi kodları okurken bu satırları okumaz.
4. Açıklama satırlarına, günlük konuşma dilinde yazdığınız gibi, istediğiniz karakteri (boşluk, Türkçe karakter vs) kullanabilirsiniz. Derleyici bu satırları görmeyeceğinden hiçbir sorun teşkil etmeyecektir.
5. Eğer açıklama satırınız birden fazla ise **/* */** şeklinde toplu açıklama satırı kullanabilirsiniz.



6.  visual studio menüsünde bulunan bu ikonlara tıklayarak da seçtiğiniz satırı açıklama satırına dönüştürebilir yada geri alabilirsiniz.

7. Bir diğer yolu da,  için **Ctrl + K , Ctrl C** ve  için **Ctrl + K , Ctrl U** kombinasyonlarını kullanmaktır.

Örnek :

1. Kullanım	2. Kullanım
<pre>#include "stdafx.h" using namespace std; void main() { // bu birinci açıklama satırıdır // bu ikinci açıklama satırıdır // bu da üçüncü açıklama satırıdır }</pre>	<pre>#include "stdafx.h" using namespace std; void main() { /* bu birinci açıklama satırıdır bu ikinci açıklama satırıdır bu da üçüncü açıklama satırıdır */ }</pre>

COUT (ÇIKIŞ) KOMUTUNUN KULLANILMASI

cout: “**C++ out**” kelimesinden türetilen türetilmiştir. Burada **out** kelimesinden anladığımız üzere bilgisayardan kullanıcıya çıkış işlemi (*karakter, sayı, metin vs. ekrana yazdırmak*) için kullanılır.

Genel kullanımı **cout<<degisken1<<degisken2 ... ;** şeklindedir.

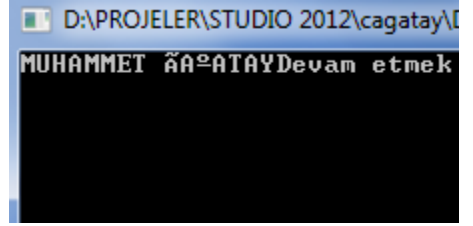
Eğer metin yazdırılacaksa değişken yerine yazdırılmak istenen metin, **cout<<"Merhaba";** örneğinde olduğu gibi “ ” **çift tırnak** arasına yazılır.

endl : İngilizcede **end of line** yani satır sonu anlamına gelmektedir. Bir alt satıra geçmek için kullanılır.

\n : Metinsel ifadeyi alt satıra geçirmek için kullanılır. Endl ile aynı işi yapar, ama metinlerin arasına da yazılabildiği için endl’ye göre daha kullanışlıdır.

\t : Metinsel ifadeler arasında 8 karakter (tab kadar) boşluk bırakmak için kullanılır.

Örnek : Ekranı isminizi yazdıran bir c++ programı yazınız.

1. Yol	2. Yol	Ekran çıktısı
<pre>#include "stdafx.h" #include <iostream> using namespace std; void main() { cout<<"MUHAMMET ÇAĞATAY"; system("pause"); }</pre>	<pre>#include "stdafx.h" #include <iostream> void main() { using std::cin; using std::cout; cout<<"MUHAMMET ÇAĞATAY"; system("pause"); }</pre>	

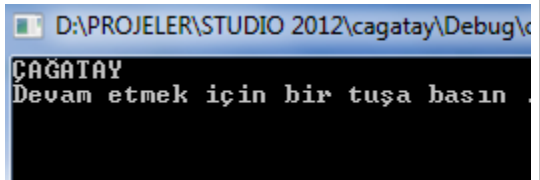
Yukarıdaki örnekte, nasıl yapıldığını görmek açısından std isim uzayını iki farklı yol ile kullandık. Bundan sonraki örneklerimizde diğerine göre daha kullanışlı bir yol olan, *using namespace std;* kodunu tek kullanacağız.

Ayrıca örnekte de görüldüğü gibi, soyadımda bulunan **ç** ve **ğ** Türkçe karakterleri, editör tarafından tanınmamaktadır ve *system("pause");* yazdığımız için görünen *devam etmek için bir tuşa basın* yazısı ile isim bitişik gözükmemektedir.

C++ editörünün Türkçe karakterleri tanıması için *#include <locale>* kütüphanesi programımıza dahil edilip *setlocale(LC_ALL,"turkish");* kodunu programımıza eklememiz gerekmektedir.

Not : Visual studio editörünün bazı versiyonlarında, *<locale>* komutu olmadanda yani sadece *<iostream>* kütüphanesinin aktiflenmesi ile de *setlocale* komutu çalışabilir.

Devam etmek için bir tuşa basın yazısını alta kaydırmak için ise *\n* yada *endl* komutlarını kullanabiliriz.

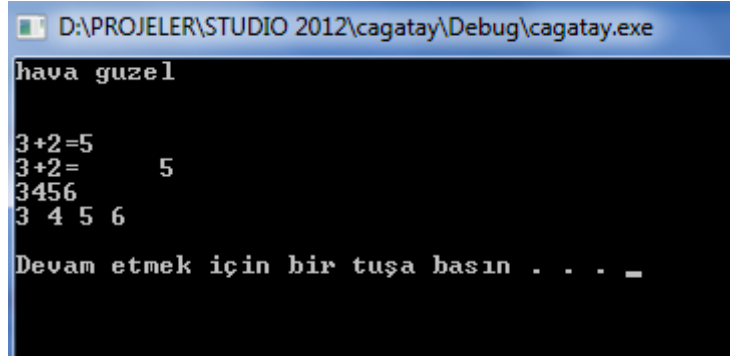
<pre>#include "stdafx.h" #include <iostream> #include <locale> using namespace std; void main() { setlocale(LC_ALL,"turkish"); cout<<"ÇAĞATAY"<<endl; system("pause"); }</pre>	<pre>#include "stdafx.h" #include <iostream> #include <locale> using namespace std; void main() { setlocale(LC_ALL,"turkish"); cout<<"ÇAĞATAY \n"; system("pause"); }</pre>	
--	---	--

Görüldüğü gibi programımız artık düzgün bir şekilde çalışıyor, ama unutmamız gereken bir şey var eğer *setlocale(LC_ALL,"turkish");* kodunu, *cout<<"ÇAĞATAY"<<endl;* kodundan sonra yazsaydık, programımızda yine Türkçe karakterler gözükmeyecekti.

Çünkü konunun başında da dediğim gibi editör kodları adım adım okumaktadır ve kodlarımızı yazarken bu çalışma mantığını göz önünde bulundurmanız gerekmektedir.

Örnek :

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main ()
{
cout<<"hava guzel"<<endl<<endl<<endl;
cout<<"3+2="<<3+2<<endl;
cout<<"3+2= \t "<<3+2<<endl;
cout<<3<<4<<5<<6<<endl;
cout<<3<<" "<<4<<" "<<5<<" "<<6<<endl;
cout<<endl;
system("pause");
}
```



Bu örnekten çıkaracağımız sonuçlar şunlar olmalıdır ;

1. **endl** komutu tek satırda birden fazla kullanılabilir ve kullanıldığı sayı kadar satır atlar.
2. **Cout<<endl;** şeklinde tek başına da kullanılabilir.
3. **cout<<3<<4<<5<<6<<endl;** kodunda olduğu gibi << den sonrakiler daima bitişik görüntülenir yani aksi yazılmadıkça yanyana yazılım söz konusudur.
Bunu önlemek için **cout<<3<<" "<<4<<" "<<5<<" "<<6<<endl;** şeklinde araya boşluklar atabiliriz.
4. **cout<<"3+2= \t "<<3+2<<endl;** de olduğu gibi **\t** komutunu metinsel ifade içinde kullanmalıyız.
5. **cout<<"3+2="<<3+2<<endl;** satırında sonuç **3+2 =5** şeklinde görüntülenmiştir. Yani cout koduna yazılan matematiksel işlemler sonuç olarak ekrana yazılırlar.

DEĞİŞKEN NEDİR ?

- **Değişken;** bilgisayarın hafızasında depolanan bilgiyi gösteren bir semboldür.
- Bu bilgiye değişkenin değeri denir.
- Atama yardımıyla değişkenin değer alması **değişken=ifade ;** şeklinde olur.
- Bu durumda öncelikle ifade hesaplanır ve değeri değişkene atanır.

DEĞİŞKEN TANIMLAMA KURALLARI

1. Değişken isimleri 32 karakter içerebilir, 32 den sonrasını editör dikkate almaz.
2. Değişken adları İngiliz alfabesinde bulunan karakterler (A-Z) veya (a-z) yada rakamlar (0-9) ile yazılmalıdır.
3. Değişken tanımlanırken Türkçe karakterler (ç, ğ, ı, ü, ö, ş) kullanılmamalıdır.
4. Değişken isimleri büyük küçük harf kullanımına duyarlıdır. Yani **orta** , **Orta** , **ORTA** , **oRTa** dördü de farklı değişkenlerdir.
5. Değişken isimleri rakam ile başlayamaz. Örnek ~~1sayi~~ yerine **sayi1** demelisiniz.
6. Değişkenlerin arasında boşluk kullanılmaz. **sayi 1** yerine **sayi_1** kullanabilirsiniz.
7. Değişken tanımlanırken alt çizgi " _ " den başka özel karakter kullanılmaz.
8. Cin, cout, min, max ... gibi C++ kodlarında var olan metinler değişken ismi olarak verilemez.

DEĞİŞKEN TIPLERİ

int : İnteger'in kısa adıdır. Tamsayı değişkenidir ve değerleri ... -3 , -2 , -1 , 0 , 1 , 2 , 2 ... biçimindedir.

float : Ondalıklı (*küsüratlı*) sayılar için kullanılan değişken tipidir, tamsayılar kümesini de kapsar. Değerleri: 0.023 , -2.3 , 4.06 , 4.0 , 6 , 7.1 ... şeklinde olabilir.

double : floattan duyarlılığı daha yüksek olan ondalık sayılar için kullanılan değişken tipidir. Değerleri : 4.0 , 6 , 7.1 , 7.6578886555 , -5.12356666 Şeklinde olabilir.

char : Bir karakter yada karakter dizisi (katar) tanımlamak için kullanılır. Her zaman bu karakterler harf olmak zorunda değildir sayılar da bir karakter olarak alınabilir. Örnek : a, A , c , C , 1 , 5 ... karakterlere atama yaparken *“tek tırnak* arasında yazılırlar. Eğer karakterin dışarıdan girilmesi isteniyorsa ve kullanıcı birden daha uzun boyut bir karakter girdi ise, editör girilen karakterlerin ilkini kabul edip değişkene atar.

String : Metinsel karakterlerdir. Tanımlandıktan sonra atama yapılırken değerler *“çift tırnak* arasında yazılır. Örnek değerler : ali , AHMET, Dursun , Çaykur Rize ...

Örnek :

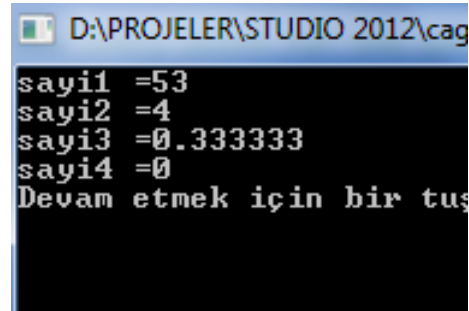
Kullanım 1	Kullanım 2
<pre>#include "stdafx.h" #include <iostream> using namespace std; void main () { int sayi1; // sayi1 adında bir tane tamsayı tanımlandı. int sayi2; float fsayi_1; float fsayi_2; double dsayi1; double dsayi2; sayi1 = 20; // sayi1 e 20 değeri atandı. sayi2 = 53; fsayi_1 = 10.322; fsayi_2 = 5.702; dsayi1=587.5457555; dsayi2=-87855.5457; system("pause"); }</pre>	<pre>void main () { int sayi1,sayi2; // sayi1 ve sayi2 adında 2 tane tamsayı tanımlandı. float fsayi_1,fsayi_2; double dsayi1,dsayi2; sayi1 = 20; sayi2 = 53; fsayi_1 = 10.322; fsayi_2= 5.702; dsayi1=587.5457555; dsayi2=-87855.5457; system("pause"); }</pre>
	Kullanım 3
	<pre>void main () { int sayi1 =20 ,sayi2=53; // sayi1 ve sayi2 adında 2 tane tamsayı tanımlanıp değerleri atandı. float fsayi_1= 10.322 ,fsayi_2=5.702; double dsayi1=587.5457555, dsayi2=87855.5457; system("pause"); }</pre>
Kullanım 4	
<pre>void main () { int sayi1 =20 ,sayi2=53; float fsayi_1= 10.322 ,fsayi_2=5.702; double dsayi1=587.5457555, dsayi2=87855.5457; system("pause"); }</pre>	

Yukarıdaki örnekten çıkarmamız gereken sonuçlar;

1. Her değişkenin bir tipi olmalıdır.
2. Değişkenler teker teker alt alta yazılarak tanımlandığı gibi, araya **virgül** koyarak ve en başta bir kez değişken tipini belirterek yan yana da yazılabilir.
3. Değişkenler önce tanımlanıp sonra değer ataması yapıldığı gibi, tanımlandıkları anda da değer ataması yapılabilirler.
4. Birden fazla değişken tipi araya **noktalı virgül** konularak aynı satırda bildirilebilir.
5. Değişkenlerin tip bildirimlerinin programın başında yapılması önerilir, istenirse programın içindede yapılabilir.
6. **Noktalı virgül**'den sonra satır devam ettirilebilir.

Örnek :

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main ()
{
    int sayi1,sayi2; double sayi3,sayi4;
    sayi1=53, sayi2=9/2, sayi3=1./3, sayi4=1/3;
    cout<<"sayi1 ="<<sayi1<<endl;
    cout<<"sayi2 ="<<sayi2<<endl;
    cout<<"sayi3 ="<<sayi3<<endl;
    cout<<"sayi4 ="<<sayi4<<endl;
    system("pause");
}
```



Sayı1 : 53 olarak gerekiyordu : çıkan sonuç doğru

Sayı2 : $9/2 = 4.5$ olarak gözükmesi gerekiyordu : sayi2 double tipinde olmayıp int yani tamsayı tipinde olduğu için, 4.5 sayısının tam kısmı olan 4 tek görünür.

Sayı3 : $1/3 = 0.33$ olarak gözükmesi gerekiyordu : çıkan sonuç doğru

Sayı4 : $1/3 = 0.33$ olarak gözükmesi gerekiyordu : sayi4 ondalık sayı olduğundan rasyonel sayılarda $(1/3)$ yerine $(1./3)$ şeklinde nokta konularak kullanılmalıdır. Aksi takdirde editör sayıyı tamsayı olarak algılar ve sonucu 0 verir.

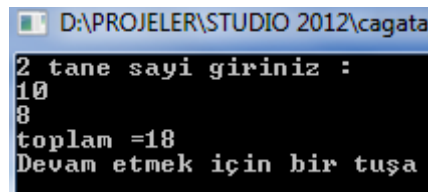
CİN (GİRİŞ) KOMUTUNUN KULLANILMASI

cin: “C++ in” kelimesinden türetilmiştir. *in*’den de anlaşılacağı üzere dışarıdan (klavyeden veya dosyadan) bilgisayara karakter, sayı, metin vs. girişi yapmak için kullanılır.

Genel kullanımı **cin>>değişken1>>değişken2 ... ;** şeklindedir.

Örnek : Kullanıcıdan iki sayı alıp bu sayıların toplamını ekrana yazdıran c++ programını yazınız.

```
void main ()
{
    int sayi1,sayi2;
    cout<<"2 tane sayi giriniz : "<<endl;
    cin>>sayi1>>sayi2;
    cout<<"toplam ="<<sayi1+sayi2<<endl;
    system("pause");
}
```



BASİT MATEMATİKSEL OPERATÖRLER

toplama	+	2+3
çıkarma	-	2-3
çarpma	*	2*3
bölme	/	2/3
parantez	()	2*(5+3)
Kalan (modüler aritmetik)	%	3%2

Not : C++ dilinde, matematikteki işlem öncelik sırası aynen geçerlidir. (yukarıdaki tabloya göre alttan üste doğudur.) ve kalan operatörü (%) sadece tamsayılarda çalışır.

MANTIKSAL OPERATÖRLER

&& : Ve operatörü	: Veya operatörü
== : Eşit midir ?	!= : Eşit değil midir ?
> : Büyük müdür ?	< : Küçük müdür ?
>= : Büyük eşit midir ?	<= : Küçük eşit midir ?

ARTIŞ AZALIŞ OPERATÖRLERİ

sayi++ ;	++sayi ;	sayi = sayi + 1 ;
sayi-- ;	--sayi ;	sayi = sayi - 1 ;

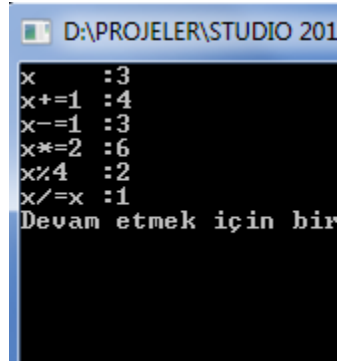
Not : sayi++ ve ++sayi bir değişkene atanırsa durum değişir. atamanın yapıldığı zaman eşittire yakın yer önce yapılır. Yani x=sayi++ da sayi değeri x ' e atanır sonra sayi değeri bir artırılır. x=++sayi da ise sayi değeri önce 1 artırılır ve sonra x ' e atanır

HIZLI İŞLEM OPERATÖRLERİ

sayi1 += sayi2 ;	sayi1 = sayi1 + sayi2 ;	sayi1 /= sayi2 ;	sayi1 = sayi1 / sayi2 ;
sayi1 -= sayi2 ;	sayi1 = sayi1 - sayi2 ;	sayi1 *= sayi2 ;	sayi1 = sayi1 * sayi2 ;

Örnek :

```
#include <iostream>
using namespace std;
void main ()
{
    int x=3;
    cout<<"x : "<<x<<endl;
    x+=1;
    cout<<"x+=1 : "<<x<<endl;
    x-=1;
    cout<<"x-=1 : "<<x<<endl;
    x*=2;
    cout<<"x*=2 : "<<x<<endl;
    x=x%4;
    cout<<"x%4 : "<<x<<endl;
    x/=x;
    cout<<"x/=x : "<<x<<endl;
    system("pause");
}
```



PROGRAMLAMAYA GİRİŞ

Algoritma kavramı : Programla dili için günlük konuşma diliyle yazılmış ve bir problemin çözümüne yönelik yazılan adımların bütününe denir.

Algoritmaların diğer adı da yalancı kodlardır ve kullandığımız Türkçe konuşma dili ile yazılıp istenilen programlama dilinde kodlanırlar. Biz de bu dersimizde algoritmalar c++ dili ile kodlayacağız.

- Algoritmalar **başla** ve **bitir** komutları arasına yazılırlar.
- **Oku** : kullanıcıdan bilgisayara giriş yapmak için yazılır.
- **Yaz** : bilgisayar tarafından ekrana çıktı vermek için kullanılır.
- **Eğer / değilse** : karşılaştırma ve koşulları belirtmek için kullanılır.
- **Git** : verilen adıma yönlendirmek için kullanılır.

Örnek : klavyeden girilen iki sayıyı çarpan programın algoritmasını yazınız e c++ dilinde kodlayınız.

Türkçe algoritma	C++ kodu
<ol style="list-style-type: none">1. Başla2. Sayısal sayı1, sayı2, sonuc3. Yaz; "İki sayı giriniz"4. Oku; sayı15. Oku; sayı26. sonuc = sayı1 * sayı27. Yaz; "Sonuç: " & sonuc8. Bitir	<pre>#include "stdafx.h" #include <iostream> using namespace std; void main () { int sayı1, sayı2 ,sonuc; cout <<"2 sayıyı giriniz"<<endl; cin >>sayı1; cin >>sayı2; sonuc= sayı1 * sayı2; cout<<"sonuç ="<<sonuc<<endl; system("pause"); }</pre>

ALGORİTMA AKIŞ DİYAGRAMLARI

Herhangi bir sorunun çözümü için izlenmesi gerekli olan aritmetik ve mantıksal adımların söz veya yazı ile anlatıldığı algoritmanın, görsel simge ya da sembollerle ifade edilmiş şekline Akış Şemaları, Akış Diyagramları veya FLOWCHART denir.

Akış şemalarının algoritmadan farkı, adımların simgeler şeklinde kutular içine yazılmış olması ve adımlar arasındaki ilişkilerin ve yönünün oklar ile gösterilmesidir.

Akış şemalarının hazırlanmasında aşağıda yer alan simgeler kullanılır.



Örnek : iki sayıyı toplayan programın algoritma akış diyagramını çiziniz.



Örnek : girilen 2 sayıda hangisinin daha büyük olduğunu bulan programın akış diyagramını çiziniz.



CMATH (MATEMATİK) KÜTÜPHENESİ

1. C++ dilinde matematiksel fonksiyonların aktiflenmesi için *cmath* kütüphanesi `#include <cmath>` şeklinde eklenmelidir.
2. Aynı kütüphane c dilinde *math.h* ile aktif edilir. *math.h* kütüphanesi c++ dilinde de çalışır.
3. Kütüphanede kullanılan sayısal değerler, *float* veya *double* tipinde tanımlanmalıdır. Tamsayı şeklinde tanımlanırsa hata verirler.
4. Matematiksel eşitlikleri yazarken parantezlere özellikle dikkat edilmelidir.

Örneğin : $(x+y)/(x-y)$ ile $x+y/x-y$ tamamen farklıdır.

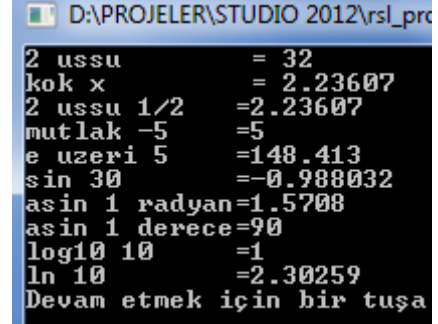
İlk değerde sonuç $\frac{(x+y)}{(x-y)}$ iken ikinci değerde sonuç $x + \frac{y}{x} - x$ çıkar.

MATEMATİKSEL FONKSİYONLAR

pow(x,y)	x üssü y demektir. <i>pow(x,0.5)=sqrt(x)</i> dir.
sqrt(x)	x in karekökü demektir.
fabs(x)	x in mutlak değeri demektir.
exp(x)	e üssü x demektir. Burada <i>e=2.71...</i> dir.
sin(x) , cos(x)	radyan olarak x sayısının sinüs ve kosinüs değerlerini verir.
tan(x) , cot(x)	radyan olarak x sayısının tanjant ve kotanjant değerlerini verir.
asin(x) , acos(x)	radyan olarak x sayısının arcsinüs ve arccosinüs değerlerini verir.
atan(x) , acot(x)	radyan olarak x sayısının arctanjant ve arckotanjant değerlerini verir.
log10(x)	10 tabanında x in logaritmasını verir.
log(x)	ln(x) anlamına gelir.

Örnek :

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main ()
{
    double x=5, y=-5 , pi;
    cout<<"2 ussu      = "<<pow(2,x)<<endl;
    cout<<"kok x      = "<<sqrt(x)<<endl;
    cout<<"2 ussu 1/2   = "<<pow(x,0.5)<<endl;
    cout<<"mutlak -5    = "<<fabs(y)<<endl;
    cout<<"e uzeri 5     = "<<exp(x)<<endl;
    // fonksiyonların içinde tam sayı kullanılmaz,
    // eğer sayı tam sayı ise sonuna nokta konulur.
    // değişken olarak double kullanılmalıdır.
    cout<<"sin 30      = "<<sin(30.)<<endl;
    cout<<"asin 1 radyan="<<asin(1.)<<endl;
    pi=3.141592654;
    cout<<"asin 1 derece="<<asin(1.)*180/pi<<endl;
    cout<<"log10 10     = "<<log10(10.)<<endl;
    cout<<"ln 10        = "<<log(10.)<<endl;
    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012\rs_l_pr
2 ussu      = 32
kok x      = 2.23607
2 ussu 1/2   = 2.23607
mutlak -5    = 5
e uzeri 5     = 148.413
sin 30      = -0.988032
asin 1 radyan=1.5708
asin 1 derece=90
log10 10     = 1
ln 10        = 2.30259
Devam etmek için bir tuşa
```

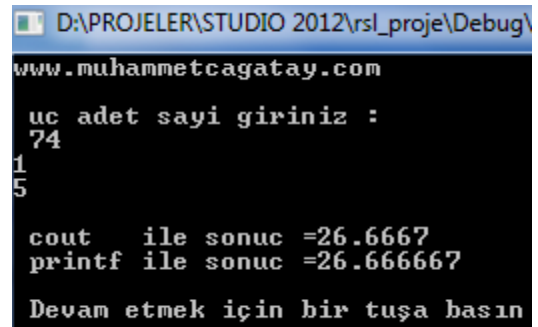
PRINTF ÇIKIŞ KOMUTUNUN KULLANIMI

Cout komutunda olduğu gibi ekrana çıktı vermek için kullanılır.

- Kullanımı : **printf** (format karakterleri, argüman listesi); şeklindedir.
- Format karakterleri önüne **%** işareti konulur.
- bu karakterler argümanın tipine göre değişirler ve Format karakterleri şunlardır ;
- **c** : karakter
- **s** : string(char dizisi)
- **d, i** : tamsayı
- **f** : double, float
- **p** : bir pointer'ın değeri

Örnek : girilen 3 sayının ortalamasını bulup printf ve cout ile iki şekilde de ekrana yazdıran program.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main ()
{
    printf("uc adet sayi giriniz : \n ");
    double sayi1,sayi2,sayi3 ,ortalama;
    cin>>sayi1>>sayi2>>sayi3;
    ortalama= (sayi1 + sayi2 +sayi3)/3;
    cout<<"\n cout   ile sonuc = "<<ortalama<<endl;
    printf(" printf ile sonuc =%f \n \n ",ortalama);
    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012\rs_l_proje\Debug\
www.muhammetcagatay.com
uc adet sayi giriniz :
74
1
5
cout   ile sonuc =26.6667
printf ile sonuc =26.666667
Devam etmek için bir tuşa basın
```

Örnek : exp sayısını farklı formatlarda ekrana yazdıran program. $\exp(1)=2.7182818284590452354...$ dir.)

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main ()
{
    float x; double y;
    x=exp(1.);
    y=exp(1.);

    cout<<"\n\n***** Cout Direk yazdirma *****"<<endl;
    cout<<exp(1.)<<endl;
    cout<<x<<endl;
    cout<<y<<endl<<endl<<endl;

    cout<<"***** printf li formatsiz yazdirma *****"<<endl;
    printf("%f \n", exp(1.));
    printf("double e=%f \t float e=%f \n \n",x,y);

    cout<<"***** printfli formatli yazdirma *****"<<endl;
    printf("%1.10f \n", exp(1.));
    //tam kısımdan 1 ondalık kısımdan 10 basamak yazdırma
    printf("%1.10f \t %1.10f \n",x,y);
    system("pause");
}
```

```
***** cout Direk yazdirma *****
2.71828
2.71828
2.71828

***** printf li formatsiz yazdirma *****
2.718282
double e=2.718282          float e=2.718282

***** printfli formatli yazdirma *****
2.7182818285
2.7182817459      2.7182818285
Devam etmek için bir tuşa basın . . . _
```

Örnek : Kullanıcı tarafından girilen x değerine göre $\frac{(x+3)e^4 - 2x + \sin x}{\pi \ln x + 5x}$ işleminin sonucunu hesaplayıp ekrana yazdıran c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main ()
{
    double x,sonuc;
    double pi_sayisi=3.14;

    printf("bir x degeri giriniz : ");
    cin>>x;
    sonuc=((x+3)*exp(4.)-2*x+sin(x))/(pi_sayisi*(log(x)+5*x));
    cout<<"sonuc = "<<sonuc<<endl;
    system("pause");
}
```

D:\PROJELER\STUDIO 2012\rs_l_proje\Debug\rs_l_proje.exe

```
bir x degeri giriniz : 54
sonuc = 3.49116
Devam etmek için bir tuşa basın . . . _
```

İF KOŞUL YAPISI

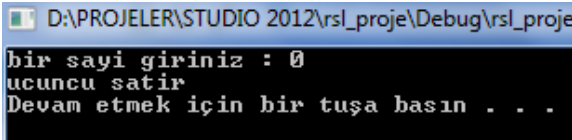
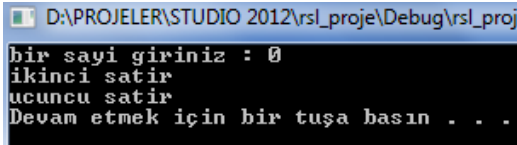
İf İngilizcede **eğer**, else de **değilse** demektir. Eğer böyle ise şöyle şöyle yap tarzındaki koşulları bilgisayara anlatmak için kullanılır.

- şart doğru ise durum 1 yanlış ise durum 2 gerçekleşecektir.
- Eğer şart yerine **0** sayısı yazılırsa veya gelirse **şart sağlanmamış**,
- Şart yerine **0 harici bir sayı** gelirse veya yazılırsa **şart sağlanmış** demektir.
- **Else** her zaman yazılmak zorunda değildir, aksi bir durum incelenmek istenirse yazılır.

Kullanım 1	Kullanım 2
<pre>if (şart) { şart salandıkça olaylar } if (şart) { şart salandıkça olaylar } else { olayların ters durumu }</pre>	<p>Eğer olaylarımız sadece 1 tane ise blok { } açmadan tek satırda da kullanılabilir.</p> <p>if (şart) şart salandıkça olay</p> <p>Veya</p> <p>if (şart) şart salandıkça olay else olayların ters durumu</p>

Not : Eğer blok yapı kullanılmazsa ve koşuldan sonra birden fazla olay yazılmış ise sadece ilk olay koşula bağlı olarak çalışır.

Aşağıdaki örnekte görüldüğü gibi blok yapı kullanılıp 0 değeri girdiğimizde **yani şart sağlanmayıp derleyici if döngüsünün içine girmediğinde**, ilk iki satırı atlayıp sadece üçüncü satırı yazdığını, blok kullanılmadan yapılan programda ise sadece ilk satırı atladığını görmek mümkündür.

<pre>#include "stdafx.h" #include <iostream> using namespace std; void main () { int sayi; cout<<"bir sayi giriniz : "; cin>>sayi; if (sayi>0) { cout<<"birinci satir"<<endl; cout<<"ikinci satir"<<endl; } cout<<"ucuncu satir"<<endl; system("pause"); }</pre> 	<pre>#include "stdafx.h" #include <iostream> using namespace std; void main () { int sayi; cout<<"bir sayi giriniz : "; cin>>sayi; if (sayi>0) { cout<<"birinci satir"<<endl; cout<<"ikinci satir"<<endl; cout<<"ucuncu satir"<<endl; system("pause"); } }</pre> 
---	---

Örnek : Klavyeden girilen 2 sayı için, birinci sayının ikinci sayıya tam bölünüp bölünmediğine bakan ve durumu ekrana yazdıran c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

void main ()
{
    int sayi1,sayi2;
    cout<<"birinci sayiyi giriniz : ";
    cin>>sayi1;
    cout<<"ikinci sayiyi giriniz : ";
    cin>>sayi2;

    if (sayi1 % sayi2)
    {
        cout<<sayi1<<" sayisi " <<sayi2<<" ye tam bolunmez."<<endl;
    }
    else
    {
        cout<<sayi1<<" sayisi " <<sayi2<<" ye tam bolunur."<<endl;
    }

    system("pause");
}
```

BLOK {} KULLANMADANYAPILMIŞ HALİ :

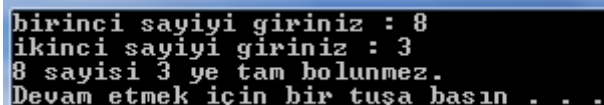
```
#include "stdafx.h"
#include <iostream>
using namespace std;

void main ()
{
    int sayi1,sayi2;
    cout<<"birinci sayiyi giriniz : ";
    cin>>sayi1;
    cout<<"ikinci sayiyi giriniz : ";
    cin>>sayi2;

    if (sayi1 % sayi2) cout<<sayi1<<" sayisi " <<sayi2<<" ye tam bolunmez."<<endl;

    else cout<<sayi1<<" sayisi " <<sayi2<<" ye tam bolunur."<<endl;

    system("pause");
}
```



```
birinci sayiyi giriniz : 8
ikinci sayiyi giriniz : 3
8 sayisi 3 ye tam bolunmez.
Devam etmek için bir tuşa basın . . .
```

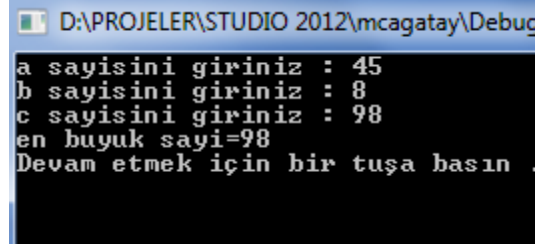
EKRAN ÇIKTISI :

Örnek : klavyeden girilen 3 sayı arasından en büyük olanı bulan c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main ()
{
    int a,b,c,enb;
    cout<<"a sayisini giriniz : ";
    cin>>a;
    cout<<"b sayisini giriniz : ";
    cin>>b;
    cout<<"c sayisini giriniz : ";
    cin>>c;
    enb=a; // ilk sayiyi olarak en buyuk
    olarak kabul ediyoruz ve digerleri ile
    kiyaslayaagiz

    if(b>enb) enb=b;
    if(c>enb) enb=c;
    cout <<"en buyuk sayi="<<enb<<endl;

    system("pause");
}
```



D:\PROJELER\STUDIO 2012\mcagatay\Debug

```
a sayisini giriniz : 45
b sayisini giriniz : 8
c sayisini giriniz : 98
en buyuk sayi=98
Devam etmek için bir tuşa basın
```

Örnek : $f(x) = \begin{cases} x^2, & x < 0 \\ x + 1, & x \geq 0 \end{cases}$ klavyeden girilen x değerine göre fonksiyonun sonucunu ekrana yazdıran c++ programını yazınız.

1. yol	2. yol
<pre>#include "stdafx.h" #include <iostream> using namespace std; void main () { double x; cout<<"x sayisini giriniz : "; cin>>x; if(x<=0) //x negatifse { } arasındaki işlemler yapılır. { x*=x; cout<<x<<endl; } else cout<<x+1<<endl; system("pause"); }</pre>	<pre>void main () { double x; cout<<"x sayisini giriniz : "; cin>>x; if(x<=0) cout<<x*x<<endl; else cout<<x+1<<endl; system("pause"); }</pre>
	3. yol
	<pre>if(x<=0) { x=x*x; cout<<x<<endl; } else { x=x+1; cout<<x<<endl; }</pre>

İF – ELSE İF KOŞUL YAPISI

Kullanım 1	Kullanım 2
<pre>if (şart 1) { olay 1 } else if (şart 2) { olay 2 } ... else { olayların hiçbirinin sağlanmadığı bir olay }</pre>	<pre>if (şart 1) olay 1 else if (şart 2) olay 2 ... else olayların hiçbirinin sağlanmadığı bir olay</pre>

Örnek : $\Delta = b^2 - 4ac$ olmak üzere $ax^2 + bx + c = 0$ denkleminin köklerini bulan c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main ()
{
    int a, b, c;
    double x1, x2, kok, delta;
    cout<<" a katsayisini giriniz"<<endl;
    cin>>a;
    cout<<" b katsayisini giriniz"<<endl;
    cin>>b;
    cout<<" c katsayisini giriniz"<<endl;
    cin>>c;
    delta = b*b-4*a*c;
    if(a==0 && b==0)
    {
        cout<<"denklemin cozumu yoktur"<<endl;
    }
    else if(a==0)
    {
        kok= -c/b;
        cout<<"denklemin koku="<<kok<<endl;
    }
    else if (delta>=0)
    {
        x1 = (-b + sqrt(delta))/(2*a);
        x2 = (-b - sqrt(delta))/(2*a);
        cout<<"birinci kok="<<x1<<"\t"<<"ikinci kok="<<x2<<endl;
    }
    else cout<<"denklem kompleks koke sahip"<<endl;
    system("pause");
}
```

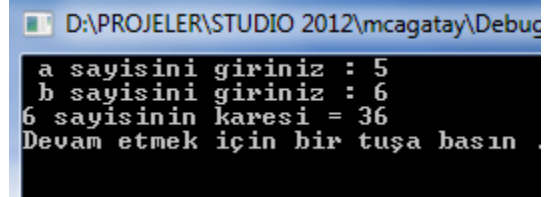
İF – ELSE İÇİN KISA KONTROL YAPISI

değişken = (şart) ? a:b

Şart sağlanırsa değişkene **a** değeri
Şart sağlanmaz ise değişkene **b** değeri atanır

Örnek : Yukarıdaki yapıyı kullanarak, klavyeden girilen 2 sayıdan büyük olanın karesini ekrana yazdıran c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main ()
{
    int a, b, c;
    cout<<" a sayisini giriniz : ";
    cin>>a;
    cout<<" b sayisini giriniz : ";
    cin>>b;
    c= a>b? a:b;    // a sayısı b'den büyükse
    c ye a'yı atar, değilse c'ye b'yi atar
    cout<<<<" sayisinin karesi = "<<c*c<<endl;
    system("pause");
}
```



VERİ TİPLERİNİ DÖNÜŞTÜRMEK

Sqrt, sin, log gibi matematik fonksiyonlar, değerleri *double* değişken olarak istedikleri için; bazen tam sayıları (int leri) double değişkene atamak zorunda kalabiliriz.

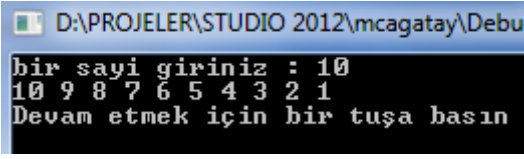
Örnek :

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main ()
{
    int x=2;
    //sqrt(x) çalışmaz. Çünkü x tamsayıdır.
    double y=x; //x tam sayı ama y=x double oldu.
    cout<<sqrt(y)<<endl;
    system("pause");
}
```


WHILE DÖNGÜSÜ

<pre>while (şart) { şart sağlandığı müddetçe işlemler }</pre>	<p>while (şart) şart sağlandığı müddetçe işlemler</p> <ul style="list-style-type: none">şart sağlandığı müddetçe durum gerçekleştirilir.while(1) sonsuz dögüdür. Çünkü 1 yerinde 0 hariç ne yazılırsa yazılsın şart hep doğru olacağından işlemler hep devam eder.
---	---

Örnek : Klavyeden girilen sayıya kadar olan bütün sayıları yan yana yazdıran c++ programını, **while** dögüsünü kullanarak yazınız.

<pre>#include "stdafx.h" #include <iostream> using namespace std; void main () { int sayi; cout<<"bir sayi giriniz : "; cin >>sayi; while (sayi) { cout<<sayi<<" "; sayi--; } cout<<endl; system("pause"); }</pre>	<p>Diğer bir yol</p> <pre>void main() { int sayi,i=0; cout<<"bir sayi giriniz : "; cin>>sayi; while(i<=sayi) { cout<<i<<" "; i++; } system("pause"); }</pre> 
---	---

DO WHILE DÖNGÜSÜ

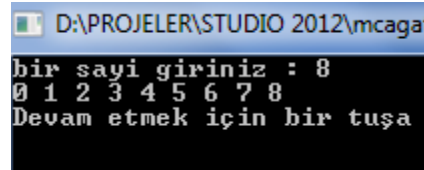
<ul style="list-style-type: none">şart sağlandığı müddetçe işlem gerçekleştirilir.do ... while ' da önce işlem yapılır sonra kontrol olur.while de ise önce kontrol yapılır sonra işlem olur.	<pre>do { şart sağlandığı müddetçe işlemler } while (şart) ;</pre>
---	--

GOTO YAPISI

<pre>goto adim; adim:</pre>	<ul style="list-style-type: none">ingilizcedeki git kelimesinden uyarlanmıştır.programı belirli bir işlemten belirli bir yere yönlendirmek için kullanılır. (<i>baş a git , son a git , adım1'e git gibi</i>)
-------------------------------------	--

Örnek : Klavyeden girilen sayıya kadar olan bütün sayıları yan yana yazdıran c++ programını, *do while* döngüsünü kullanarak yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int sayi,i=0;
    cout<<"bir sayi giriniz : ";
    cin>>sayi;
    do
    {
        cout<<i<<" ";
        i++;
    }
    while(i<=sayi);
    cout<<endl;
    system("pause");
}
```



Örnek : Klavyeden girilen sayıya kadar olan bütün sayıları alt alta yazdıran c++ programını, *goto* yapısını kullanarak yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int sayi,i=0;
    cout<<"sayi gir : ";
    cin>>sayi;

    bas:
    cout<<i<<" "<<endl;
    i++;

    if (i<=sayi) goto bas;

    system("pause");
}
```

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int sayi,i=0;
    cout<<" sayi gir : ";
    cin>>sayi;

    bas:
    cout<<i<<" "<<endl;

    if (i<sayi)
    {
        i++;
        goto bas;
    }

    system("pause");
}
```



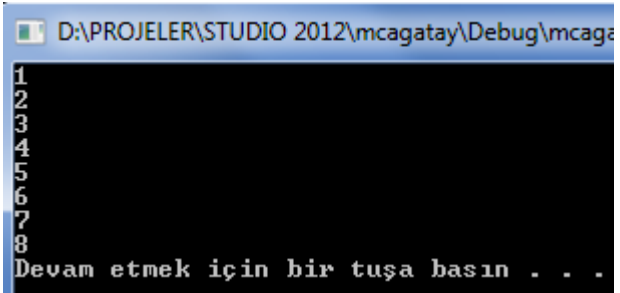
Not : Yukarıda verdiğimiz **bas:** bir c++ kodu değildir, **"başa git"** emrini vermek için tanımlanan bir değişkendir.

FOR DÖNGÜSÜ

Kullanım 1:	Kullanım 2 :
<pre>for (int değişken = başlangıç ; şart ; arttırım) { olaylar... }</pre>	<pre>int değişken; for (değişken = başlangıç ; şart ; arttırım) { olaylar... }</pre>

1. Verilen başlangıç değerinden biti değerine kadar saydırıp işlem yaptırmak için kullanılan döngü tipidir.
2. önce başlangıç değeri işleme alınır ve sonra artış miktarına göre işlemler bitiş şartı sağlandığı sürece devam eder.
3. **for(;;)** sonsuz döngü anlamına gelir.
4. Artım miktarı hep tam sayı olmak zorunda değildir. **for(i=1;i<=n;i+=0.1)** denildiğinde i nin değeri her adımda 0.1 artacaktır. Dikkat edilmesi gereken nokta i nin artık **int** olmadığı **float** veya **double** olması gerektiğidir.
5. Döngü sadece artarak değil **for(i=n;i>=0;i--)** kodu ile azalarak da sayabilmektedir.

Örnek : indexi birden başlayıp, index dokuzdan küçük oluncaya kadar artan, ve ekrana index değerini ekrana yazan c++ programını yazınız. (yani 1 den 9 a kadar olan sayıları ekrana yazdıran program)

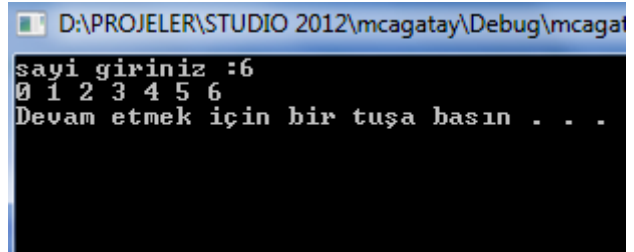
KULANIM 1	EKRAN ÇIKTISI
<pre>#include "stdafx.h" #include <iostream> using namespace std; void main() { for (int i = 1; i < 9; i++) { cout<<i<<endl; } system("pause"); }</pre>	
KULLANIM 2	KULLANIM 3
<pre>void main() { for (int i = 1; i < 9; i++) cout<<i<<endl; system("pause"); }</pre>	<pre>void main() { int i for (i = 1; i < 9; i++) cout<<i<<endl; system("pause"); }</pre>

Not : Artış miktarı bir ise **i++** yazmak yeterlidir. Yukarıdaki örnekte, Eğer sayı ikişerli artsaydı **i=i+2** yazacaktık.

Örnek : Kullanıcının gireceği herhangi bir pozitif tamsayıya kadarki sayıları yanyana birer boşlukla yazan c++ programını for döngüsü kullanarak yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int sayi;
    cout<<"sayi giriniz :";
    cin>>sayi;
    for(int i=0;i<=sayi;i++)
        cout<<i<<" ";

    cout<<endl;
    system("pause");
}
```



D:\PROJELER\STUDIO 2012\mcagatay\Debug\mcagatay.exe

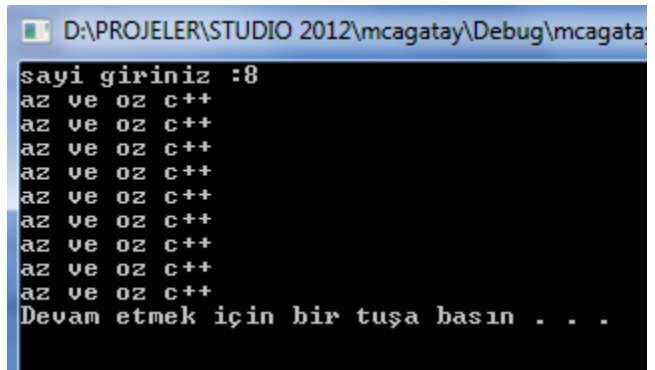
```
sayi giriniz :6
0 1 2 3 4 5 6
Devam etmek için bir tuşa basın . . .
```

Örnek : Kullanıcının gireceği sayı kadar, ekrana alt alta "az ve oz c++" yazan c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int sayi;
    cout<<"sayi giriniz :";
    cin>>sayi;

    for(int i=0;i<=sayi;i++)
        cout<<"az ve oz c++"<<endl;

    system("pause");
}
```



D:\PROJELER\STUDIO 2012\mcagatay\Debug\mcagatay.exe

```
sayi giriniz :8
az ve oz c++
az ve oz c++
az ve oz c++
az ve oz c++
az ve oz c++
az ve oz c++
az ve oz c++
az ve oz c++
Devam etmek için bir tuşa basın . . .
```

BREAK DEYİMİ : sadece for döngüsünde değil, bütün döngülerde kullanılan bir kod olup, kullanıldığı yapının içinden (döngüden) çıkmayı sağlar.

CONTINUE DEYİMİ : Döngünün bir sonraki tura atlamasını sağlar,normalde döngüler blok içerisindeki kodların tümünü okuyup bir sonraki tura geçerlerken, continue kullanıldığında; continue nin altına yazılan kodlar derleyici tarafından o tur sırasında okunmaz ve bir sonraki tura geçilir.

DÖNGÜLER İLE İLGİLİ ÖRNEK ÇÖZÜMLERİ :

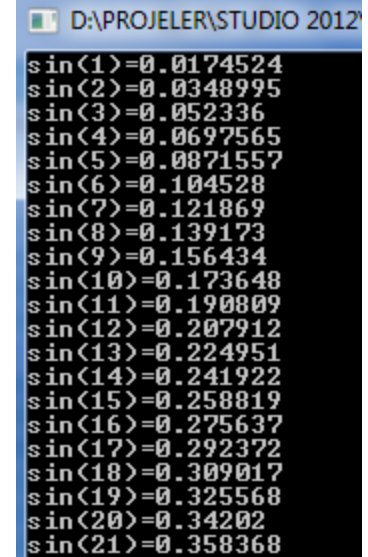
Örnek : 1 den 90'a kadar olan bütün sin x değerlerini bulup görüntüleyen c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main()
{
    double pi=3.141592654;

    for (int i=1;i<=90;i++)
    {
        double x=i;
        cout<<"sin("<<i<<"")="<<sin(x*pi/180)<<endl;

    }

    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012'
sin(1)=0.0174524
sin(2)=0.0348995
sin(3)=0.052336
sin(4)=0.0697565
sin(5)=0.0871557
sin(6)=0.104528
sin(7)=0.121869
sin(8)=0.139173
sin(9)=0.156434
sin(10)=0.173648
sin(11)=0.190809
sin(12)=0.207912
sin(13)=0.224951
sin(14)=0.241922
sin(15)=0.258819
sin(16)=0.275637
sin(17)=0.292372
sin(18)=0.309017
sin(19)=0.325568
sin(20)=0.34202
sin(21)=0.358368
```

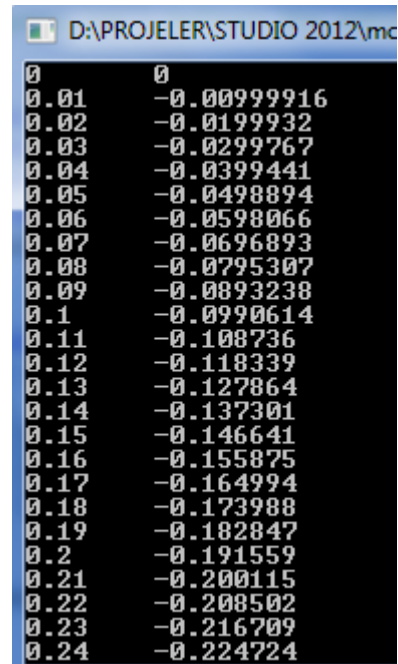
Örnek : $f(x) = e^x x^3 - 2x + \sin x$ fonksiyonunun değerlerini, $x=0, 0.01, 0.02, \dots, 9.99, 10 \dots$ hesaplayan ve bulunan. fonksiyonun değeri $1.2 < f(x) < 1.3$ olduğunda işlemi sonlandırarak bu şartı sağlayan ilk x değerini ekrana yazdıran c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main()
{
    double f,x;
    for (x=0;x<=10;x+=0.01)
    {
        f=exp(x)*pow(x,3.)-2*x+sin(x);
        cout<<x<<"\t"<<f<<endl;

        if(f>1.2 && f<1.3) break;

    }

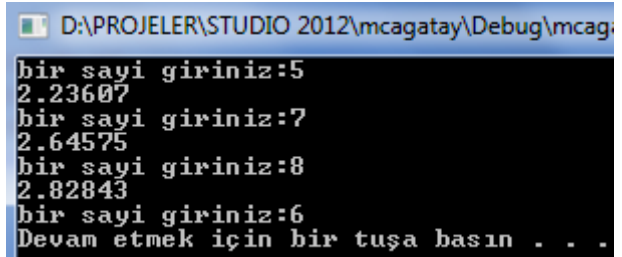
    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012\mc
0      0
0.01   -0.009999916
0.02   -0.0199932
0.03   -0.0299767
0.04   -0.0399441
0.05   -0.0498894
0.06   -0.0598066
0.07   -0.0696893
0.08   -0.0795307
0.09   -0.0893238
0.1    -0.0990614
0.11   -0.108736
0.12   -0.118339
0.13   -0.127864
0.14   -0.137301
0.15   -0.146641
0.16   -0.155875
0.17   -0.164994
0.18   -0.173988
0.19   -0.182847
0.2    -0.191559
0.21   -0.200115
0.22   -0.208502
0.23   -0.216709
0.24   -0.224724
```

Örnek : klavyeden 6 ya bölünen veya negatif olan bir sayı girilene kadar, girilen sayının karekökünü ekrana yazdıran c++ programını **while** ile yazınız.

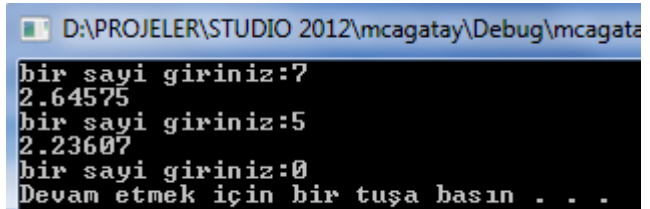
```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main()
{
    int x;
    cout<<"bir sayi giriniz:";
    cin>>x;
    while(x>=0 && (x%6)!=0)
    {
        double y=x;
        cout<<sqrt(y)<<endl;
        cout<<"bir sayi giriniz:";
        cin>>x;
    }
    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012\mcagatay\Debug\mcagatay.exe
bir sayi giriniz:5
2.23607
bir sayi giriniz:7
2.64575
bir sayi giriniz:8
2.82843
bir sayi giriniz:6
Devam etmek için bir tuşa basın . . .
```

Örnek : klavyeden 6 ya bölünen veya negatif olan bir sayı girilene kadar, girilen sayının karekökünü ekrana yazdıran c++ programını **do while** ile yazınız.

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main()
{
    int x;
    cout<<"bir sayi giriniz:";
    cin>>x;
    do
    {
        double y=x;
        cout<<sqrt(y)<<endl;
        cout<<"bir sayi giriniz:";
        cin>>x;
    }
    while(x>=0 && (x%6)!=0);
    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012\mcagatay\Debug\mcagatay.exe
bir sayi giriniz:7
2.64575
bir sayi giriniz:5
2.23607
bir sayi giriniz:0
Devam etmek için bir tuşa basın . . .
```

Örnek : klavyeden 6 ya bölünen veya negatif olan bir sayı girilene kadar, girilen sayının karekökünü ekrana yazdıran c++ programını **for döngüsü** ile yazınız.

```
#include "stdafx.h"
#include <iostream>
#include <cmath>
using namespace std;
void main()
{
    int x;
    cout<<"bir sayi giriniz:";
    cin>>x;
    if(x<=0 || (x%6)==0) goto son;
    //sart saglanırsa program son ile baslayan satırdan devam eder

    for (;;)// sonsuz dongu oluyor
    {
        double y=x;
        cout<<sqrt(y)<<endl;
        cout<<"bir sayi giriniz:"<<endl;
        cin>>x;
        if(x<=0 || (x%6)==0) break;
    }
    son:
    cout<<"girilen sayi negatif veya 6 ya bolunuyor."<<endl;
    system("pause");
}
```

Örnek : 1 den başlayıp, kullanıcının gireceği pozitif n sayısına kadar olan sayıların toplamını bulan c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int n, toplam=0;
    cout<<"n sayisi giriniz : ";
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        toplam+=i;
    }

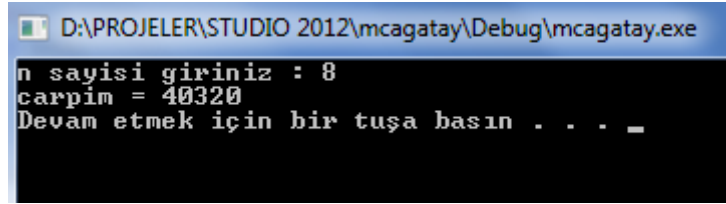
    cout<<"toplam = "<<toplam<<endl;
    system("pause");
}
```

Örnek : 1 den başlayıp, kullanıcının gireceği pozitif n sayısına kadar olan sayıların çarpımını (faktöriyelini) bulan c++ programını **for döngüsü** ile yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int n, carpim=1;
    cout<<"n sayisi giriniz : ";
    cin>>n;

    for(int i=1;i<=n;i++)
    {
        carpim*=i;
    }

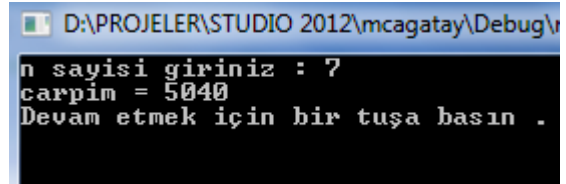
    cout<<"carpim ="<<carpim<<endl;
    system("pause");
}
```



Örnek : 1 den başlayıp, kullanıcının gireceği pozitif n sayısına kadar olan sayıların çarpımını (faktöriyelini) bulan c++ programını **while döngüsü** ile yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int n, carpim=1, sayac=1;
    cout<<"n sayisi giriniz : ";
    cin>>n;
    while (sayac<=n)
    {
        carpim*=sayac;
        sayac++;
    }

    cout<<"carpim = "<<carpim<<endl;
    system("pause");
}
```



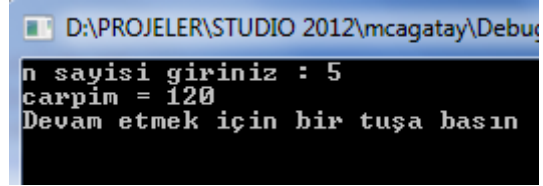
Örnek : 1 den başlayıp, kullanıcının gireceği pozitif n sayısına kadar olan sayıların çarpımını (faktöriyelini) bulan c++ programını **goto yapısı** ile yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int n, carpim=1,sayac=1;
    cout<<"n sayisi giriniz : ";
    cin>>n;

    islem:
    carpim*=sayac;
    sayac++;

    if(sayac<=n) goto islem;

    cout<<"carpim = "<<carpim<<endl;
    system("pause");
}
```



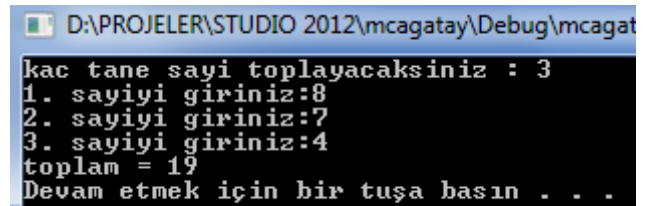
```
D:\PROJELER\STUDIO 2012\mcagatay\Debug\mcagatay.exe
n sayisi giriniz : 5
carpim = 120
Devam etmek için bir tuşa basın
```

Örnek : klavyeden girilen n tane sayının toplamını bulan C++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int adet,toplam=0,sayi;
    cout<<"kac tane sayi toplayacaksınız : ";
    cin>>adet;

    for (int i = 1; i <= adet; i++)
    {
        cout<<i<<". sayiyi giriniz:";
        cin>>sayi;
        toplam=toplam+sayi;
    }

    cout<<"toplam = "<<toplam<<endl;
    system("pause")
}
```

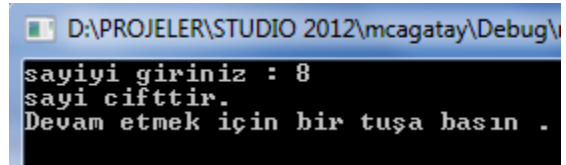


```
D:\PROJELER\STUDIO 2012\mcagatay\Debug\mcagatay.exe
kac tane sayi toplayacaksınız : 3
1. sayiyi giriniz:8
2. sayiyi giriniz:7
3. sayiyi giriniz:4
toplam = 19
Devam etmek için bir tuşa basın . . .
```

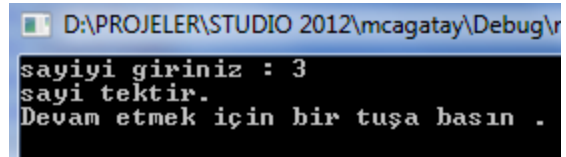
Örnek : klavyeden girilen sayının çift olup olmadığını ekrana yazdıran c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int sayi;
    cout<<"sayiyi giriniz : ";
    cin>>sayi;
    if (sayi%2==1)
        cout<<"sayi tekdir."<<endl;
    else
        cout<<"sayi çifttir."<<endl;

    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012\mcagatay\Debug\
sayiyi giriniz : 8
sayi çifttir.
Devam etmek için bir tuşa basın .
```



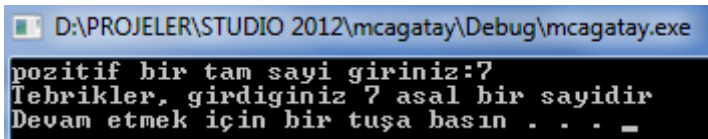
```
D:\PROJELER\STUDIO 2012\mcagatay\Debug\
sayiyi giriniz : 3
sayi tekdir.
Devam etmek için bir tuşa basın .
```

Örnek : Klavyeden girilen bir pozitif tam sayının asal olup olmadığını bulan c++ programını yazınız.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main()
{
    int i,n;
    bas:
    cout<<"pozitif bir tam sayi giriniz:";
    cin>>n;
    if(n<2)
    {
        cout<<"2 den büyük bir tam sayi giriniz"<<endl;
        goto bas;
    }

    for(i=2;i<n;i++)
        if(n%i==0) break;
    if(i==n)
        cout<<"Tebrikler, girdiginiz " <<n<<" asal bir sayidir"<<endl;
    else
        cout<<"Girdiginiz " <<n<<" asal bir sayi degildir"<<endl;

    system("pause");
}
```



```
D:\PROJELER\STUDIO 2012\mcagatay\Debug\mcagatay.exe
pozitif bir tam sayi giriniz:7
Tebrikler, girdiginiz 7 asal bir sayidir
Devam etmek için bir tuşa basın . . . _
```

SWİTCH CASE KONTROL YAPISI

1. Else if yapısı ile aynı vazifeyi görür. Yani şartlı dallanma kontrolüdür.
2. Eğer alternatif durumlar çoksa else if yerine switch case yapısını kullanmak daha mantıklıdır.

Örneğin: sayı 5 , 10 , 12 veya 20 ise kontrolü için:

İf yapısında :

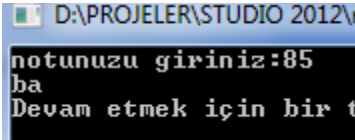
İf (sayı==5 || sayı ==10 || sayı==12 || sayı==20)
demek gerekirken

Switch yapısında :

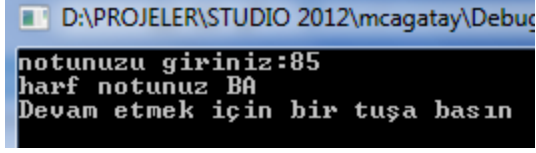
Case { 5, 10, 12 ,20 } demek yeterlidir.

```
switch (şart)
{
    case durum 1:
        olay 1
        break;
    case durum 2:
        olay 2
        break;
    ...
    default:
        varsayılan olay
        break;
}
```

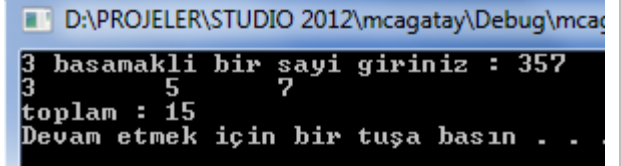
Örnek : klavyeden girilen 100 lük sistemdeki notu harf notuna çeviren c++ programını yazınız.

Kullanım 1	Kullanım 2
<pre>#include "stdafx.h" #include <iostream> using namespace std; void main() { int not; bas: cout<<"notunuzu giriniz:"; cin>>not; if(not>100 not<0) { cout<<"yanlış not girişi yaptınız\n"; goto bas; } switch (not/10) { case 9: cout<<"aa"<<endl; break; case 8: cout<<"ba"<<endl; break; case 7: cout<<"bb"<<endl; break; case 6: cout<<"cb"<<endl; break; case 5: cout<<"cc"<<endl; break; default: cout<<"ff"<<endl; break; } system("pause"); }</pre>	<pre>switch (not/10) { case 9: cout<<"aa"<<endl; break; case 8: cout<<"ba"<<endl; break; case 7: cout<<"bb"<<endl; break; case 6: cout<<"cb"<<endl; break; case 5: cout<<"cc"<<endl; break; default: cout<<"ff"<<endl; break; }</pre> 

Örnek : yukarıdaki programı bir de if yapısı ile yazalım

Kullanım2	Kullanım2
<pre>#include "stdafx.h" #include <iostream> using namespace std; void main() { int not; bas: cout<<"notunuzu giriniz:"; cin>>not; if(not>100) cout<<"not yanlis girildi"<<endl; else if(not>=90) cout<<"harf notunuz AA"<<endl; else if(not>=80) cout<<"harf notunuz BA"<<endl; else if(not>=70) cout<<"harf notunuz BB"<<endl; else if(not>=60) cout<<"harf notunuz CB"<<endl; else if(not>=50) cout<<"harf notunuz CC"<<endl; else if (not>=0) cout<<"harf notunuz FF"<<endl; else cout<<"not yanlis girildi"<<endl; system("pause"); }</pre>	<pre>#include "stdafx.h" #include <iostream> using namespace std; void main() { int not; bas: cout<<"notunuzu giriniz:"; cin>>not; if(not>100) cout<<"not yanlis girildi"<<endl; else if(not>=90) cout<<"harf notunuz AA"<<endl; else if(not>=80) cout<<"harf notunuz BA"<<endl; else if(not>=70) cout<<"harf notunuz BB"<<endl; else if(not>=60) cout<<"harf notunuz CB"<<endl; else if(not>=50) cout<<"harf notunuz CC"<<endl; else if (not>=0) cout<<"harf notunuz FF"<<endl; else cout<<"not yanlis girildi"<<endl; system("pause"); }</pre> 

Örnek : herhangi bir sayının basamak değerleri toplamını bulan c++ programını yazınız.

<pre>#include "stdafx.h" #include <iostream> using namespace std; void main() { int sayi,birler,onlar,yuzler; bas: cout<<"3 basamakli bir sayi giriniz : "; cin>>sayi; if(100>sayi 999<sayi) goto bas; yuzler=sayi/100; sayi=sayi-100*yuzler; onlar=sayi/10; birler=sayi-10*onlar; cout<<yuzler<<"\t "<<onlar<<"\t"<<birler<<endl; cout<<"toplam : "<<birler+onlar+yuzler<<endl; system("pause"); }</pre>	
---	--

FONKSİYONLAR (ALT PROGRAMLAR)

Fonksiyon : belirli sayıda verileri kullanarak bunları işleyen ve bir sonuç üreten komutlar grubu olarak adlandırılabilir. Alt program, metod, prosedür olarak da adlandırılırlar.

Bilgisayar programlama da fonksiyonlar veya alt programlar önemli yer tutmaktadır.

Bazı işlemlerin ana programın dışında yapılması ve gerektiğinde ana programda Kullanılması hem kodların kısalmasında hem de programın daha hızlı çalışmasında avantaj sağlamaktadır.

Fonksiyonlar ana programdan önce ve ana programdan sonra olmak üzere iki farklı şekilde yapılabilir. Fakat ana programdan sonra yazılan fonksiyonların örneği, ana programdan önce bildirilmelidir.

Not : Buradaki ana programdan kasıt **main** metodudur.

Fonksiyonların genel yapısı aşağıdaki gibidir :

```
FonksiyonTipi FonksiyonAdı(değişken_türü1 ad1, değişken_türü2 ad2,... )
{
    işlemler
    return geri dönüş değeri;
}
```

GERİ DÖNÜŞ TİPLERİ :

İnt : fonksiyon tamsayı döndürecek kullanılır.

Float yada **double :** fonksiyon ondalıklı sayı döndürecek kullanılır.

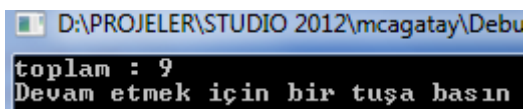
Char : fonksiyon katek harften oluşan bir rakter döndürecek kullanılır.

String : fonksiyon bir metin döndürecek kullanılır.

Void : fonksiyon geriye hiçbirşey döndürmeyecekse kullanılır.

Örnek : $f(x,y) = x + y$ (iki sayının toplamını bulan) c++ fonksiyonunu yazınız.

Ana programdan önce	Ana programdan sonra (örnek bildirerek)
<pre>#include "stdafx.h" #include <iostream> using namespace std; int topla(int x, int y) { return x+y; } void main() { cout<<topla(3,6)<<endl; system("pause"); }</pre>	<pre>#include "stdafx.h" #include <iostream> using namespace std; int topla(int,int); void main() { cout<<topla(3,6)<<endl; system("pause"); } int topla(int x, int y) { return x+y; }</pre>



Örnek : klavyeden girilen 2 sayının birbirine oranını veren c++ fonksiyonunu yazınız.

KULLANIM 1	KULLANIM 2
<pre>#include "stdafx.h" #include <iostream> using namespace std; double bol(double a, double b) { return a/b; } void main() { cout<<"2 adet sayi giriniz: "; double sayi1,sayi2; cin>>sayi1>>sayi2; cout<<"oran : "<<bol(sayi1,sayi2)<<endl; system("pause"); }</pre>	<pre>#include "stdafx.h" #include <iostream> using namespace std; double bol(double a, double b) { double bolum = a/b; return bolum; } void main() { cout<<"2 adet sayi giriniz: "; double sayi1,sayi2; cin>>sayi1>>sayi2; cout<<"oran : "<<bol(sayi1,sayi2)<<endl; system("pause"); }</pre>

Parametre : fonksiyon tanımlanırken parantez içine yazılan değişken adlarına denir.

`double bol(double a, double b)` buradaki a ve b, bol fonksiyonunun parametreleridir.

Argüman : fonksiyon çağrılırken parantez içine yazılan değişkenlerdir. `bol(sayi1,sayi2);` fonksiyonu çağırarak için kullanılan koddur ve buradaki sayi1 ve sayi2 fonksiyonun argümanlarıdır.

Yukarıdaki örnekte görüldüğü gibi argüman ve parametreler aynı değeri taşırlar fakat isimleri aynı olmak zorunda değildir.

Aynı zamanda

```
double bol(double a, double b)
{
    return a/b;
}
```

```
2 adet sayi giriniz: 8
6
oran : 1.33333
Devam etmek için bir tuşa basın
```

Fonksiyonunun parametrelerini int olarak verseydik sonuç tamsayı olarak çıkacaktı ve bizim için sağlıklı bir sonuç olmayacaktı.

```
double bol(int a, int b)
{
    return a/b;
}
```

```
2 adet sayi giriniz: 8
6
oran : 1
Devam etmek için bir tuşa basın
```

