

## Değişken Tipleri

### Tam Sayı Değişken Tipi

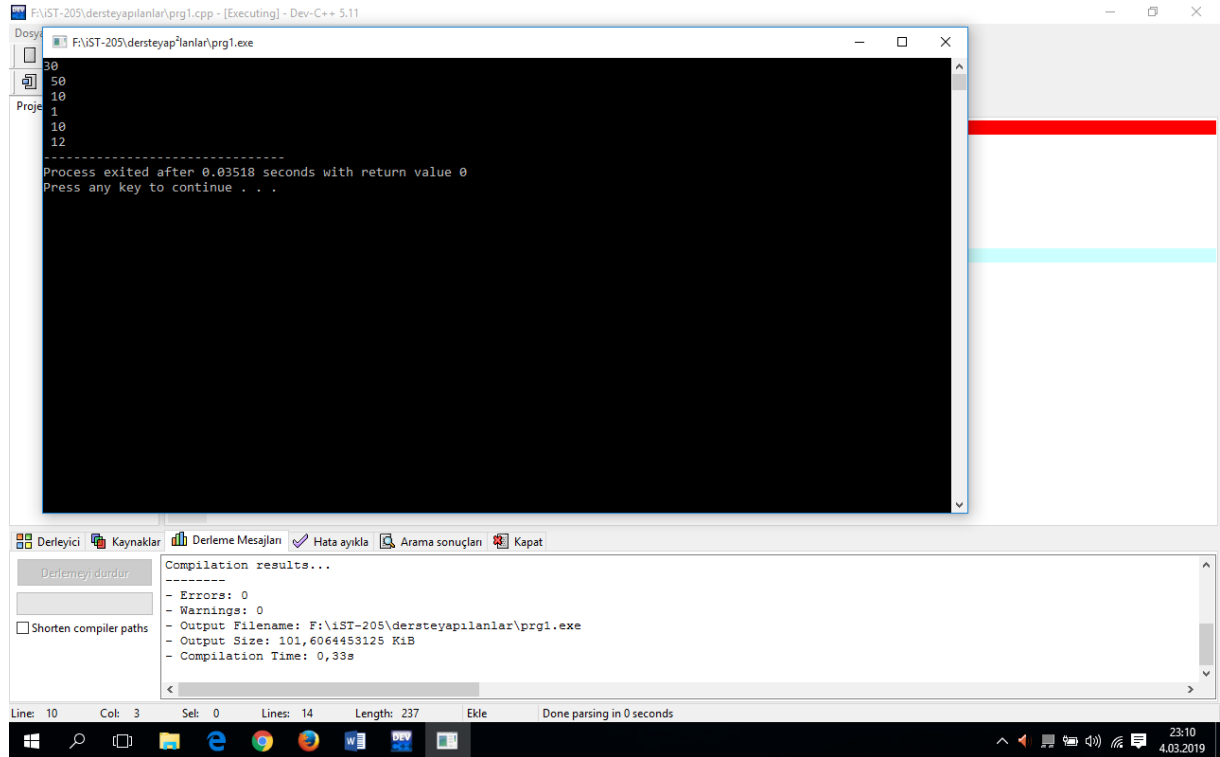
- ✓ Tam sayı tipindeki sayısal değerleri saklamak için kullanılır.
- ✓ Değişken tanımlaması 'int' kelimesi ile yapılır.
- ✓ Bu veri tipi bellekte 2, 4 ya da 8 byte boyutunda yer tutar. Bu boyut işlemciye göre değişir.
- ✓ -2.147.483.648, +2.147.483.647 aralığında değerler alabilir. Eğer bu sınırdan daha büyük ya da daha küçük değerlere ihtiyaç duyulursa long int (8 byte yer ayırır) ya da short int (2 byte yer ayırır) kullanılabilir.

int veri tipi ile kayan noktalı (ondalıklı değer) bir değer saklanmak istendiğinde sadece tam sayı kısmı saklanabilir.

### Örnek:

```
#include <stdio.h>
int main()
{
    int a=10;
    int b=20;
    int c=30;
    printf("%d",a+b);
    printf("\n %d",a+b*2);
    printf("\n %d",a*b);
    printf("\n %d",a<b);
    printf("\n %d",a++);
    printf("\n %d",++a);
}
```

programı yazılıp çalıştırıldığında aşağıdaki çıktı ekranı ile karşılaşılır.

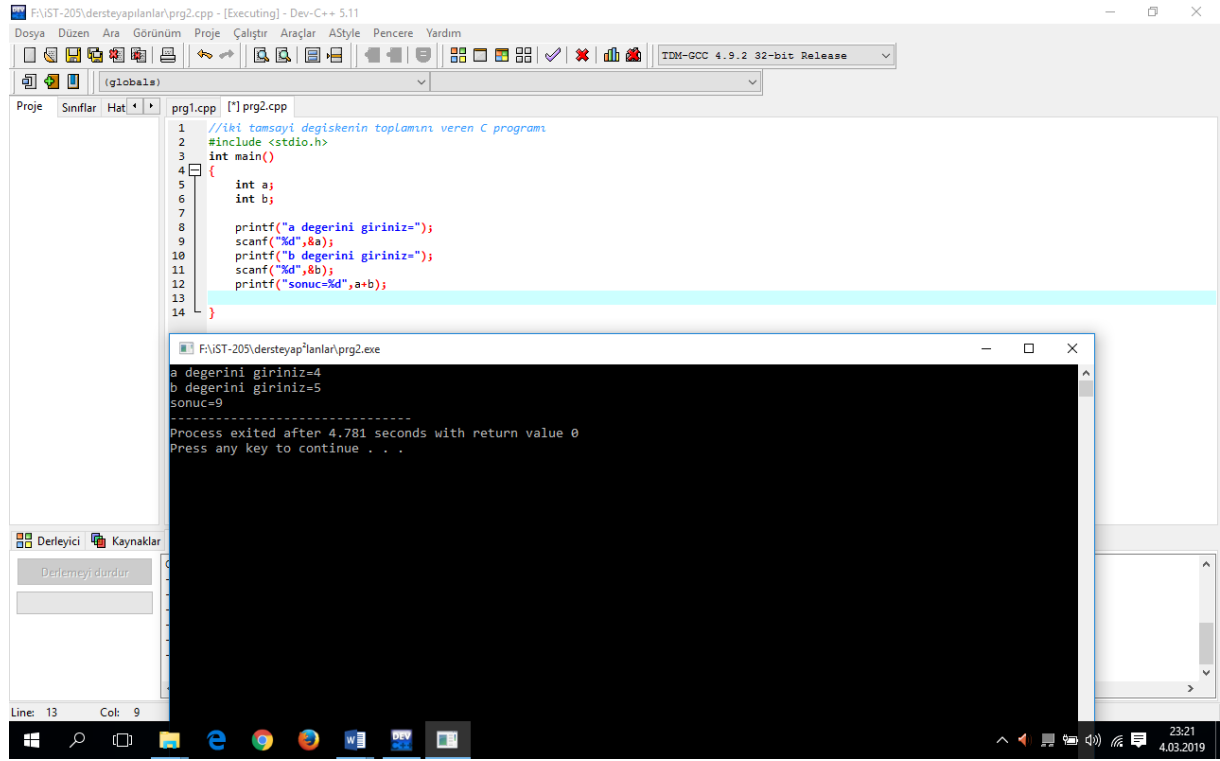


```
F:\IST-205\dersteyapilanlar\prg1.exe - [Executing] - Dev-C++ 5.11
Dosya F:\IST-205\dersteyapilanlar\prg1.exe
30
50
10
1
10
12
-----
Process exited after 0.03518 seconds with return value 0
Press any key to continue . . .

Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: F:\IST-205\dersteyapilanlar\prg1.exe
- Output Size: 101,6064453125 KiB
- Compilation Time: 0,33s

Line: 10 Col: 3 Sel: 0 Lines: 14 Length: 237 Ekle Done parsing in 0 seconds
23:10
4.03.2019
```

**Örnek:** İki tamsayı değişkeninin değerinin ekrandan okutulması ve toplamlarının yazılması programı aşağıdaki gibidir.



```
1 //iki tamsayı degişkenin toplamını veren C programı
2 #include <stdio.h>
3 int main()
4 {
5     int a;
6     int b;
7
8     printf("a degerini giriniz=");
9     scanf("%d",&a);
10    printf("b degerini giriniz=");
11    scanf("%d",&b);
12    printf("sonuc=%d",a+b);
13
14 }
```

```
a degerini giriniz=4
b degerini giriniz=5
sonuc=9
Process exited after 4.781 seconds with return value 0
Press any key to continue . . .
```

Örn; 22.57 değeri int x=22.57 diye tanımlırsa x=22 olarak algılanır. Bu durumda float ya da double değişken tipleri kullanılmalıdır.

### **Karakter Değişken Tipi**

- ✓ Karakter veri tipi sadece bir karakteri saklamak için kullanılır
- ✓ Bu veri tipi bellekte 1 byte yer tutar ve sadece 1 harf saklar.
- ✓ 'char' kelimesi ile değişken tanımlaması yapılır.

Örneğin, A harfi bellekte saklanmak istendiğinde char harf = 'A'; şeklinde saklanır.

### **Kayan Noktalı (Ondalık Sayı) Değişken Tipi**

Kayan noktalı değişken tipi 2 biçimde ifade edilebilir. Bunlar,

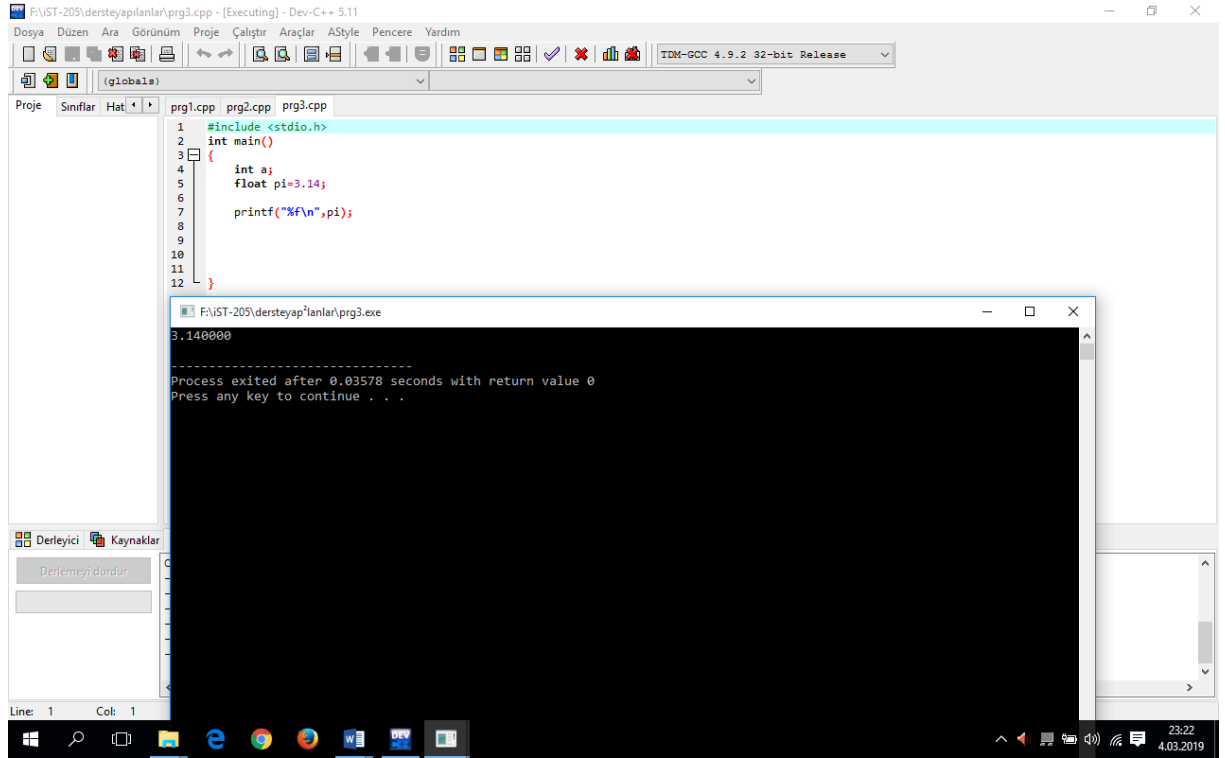
- ✓ float
- ✓ double

değişken tipleridir.

### **float Değişken Tipi**

- ✓ float veri tipi ondalıklı değerleri saklamak için kullanılır
- ✓ Bellekte 4 byte yer tutar(Bu değer int ile aynı ve işlemcilere bağlıdır).
- ✓ Bu veri tipi ile ondalık değer olarak en fazla 6 basamak kullanılır. Örn; 21.654321

## Örnek:



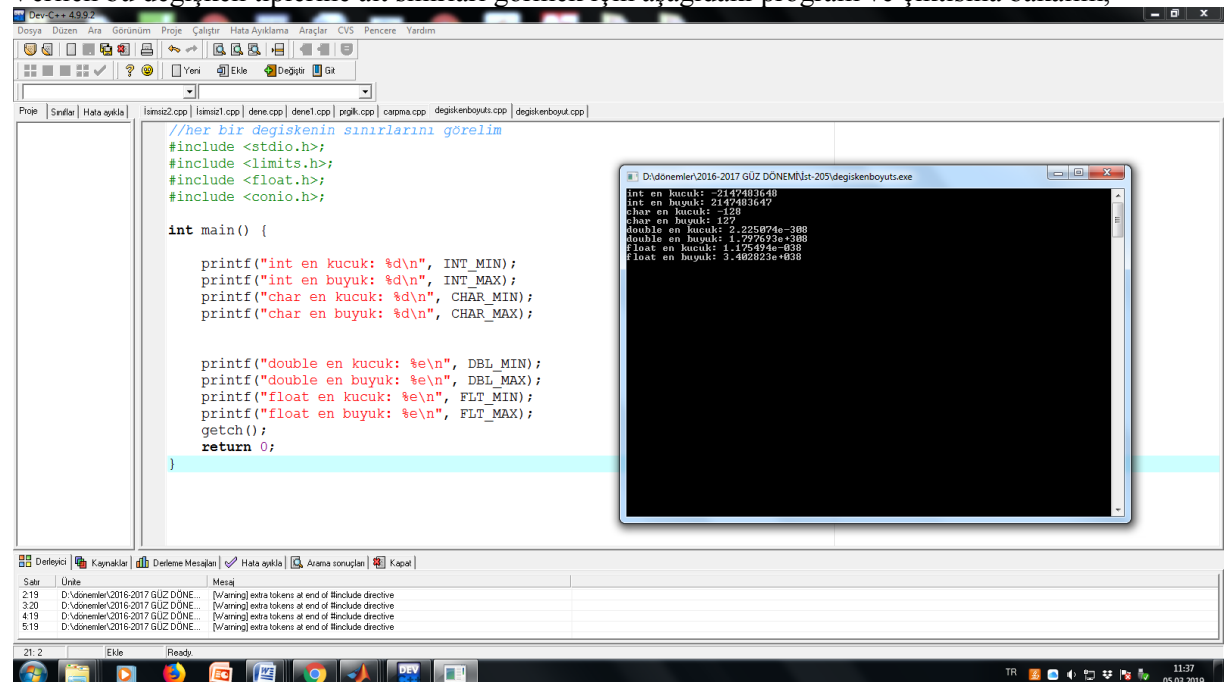
```
F:\IST-205\dersteyapilanlar\prg3.cpp - [Executing] - Dev-C++ 5.11
Dosya Düzen Ara Görünüm Proje Çalıştır Araçlar AStyle Pencere Yardım
TDM-GCC 4.9.2 32-bit Release
(globals)
Proje Simflar Hatı
prg1.cpp prg2.cpp prg3.cpp
1 #include <stdio.h>
2 int main()
3 {
4     int a;
5     float pi=3.14;
6
7     printf("%f\n",pi);
8
9
10
11
12 }
F:\IST-205\dersteyapilanlar\prg3.exe
3.140000
-----
Process exited after 0.03578 seconds with return value 0
Press any key to continue . . .
```

## double Değişken Tipi

- ✓ double veri tipinin float veri tipinden en önemli farklarından birisi ondalık değer olarak 15 basamak kullanılabilir olmasıdır. Örn; 10.123456789012345
- ✓  $3E-308$ ,  $1.7E+308$  aralığında değer alabilir ( $E-308=10^{-308}$ ,  $E+308=10^{+308}$ ).

Daha büyük bir değer saklanmak istendiğinde long double (10 byte yer ayırır) kullanılabilir.

Verilen bu değişken tiplerine ait sınırları görmek için aşağıdaki program ve çıktısına bakalım,



```
Dev-C++ 4.9.2
Dosya Düzen Ara Görünüm Proje Çalıştır Hata Ayıklama Araçlar CVS Pencere Yardım
Ekle Yeni Ekle Değiştir Git
Proje Simflar Hata ayıkla isimci2.cpp isimci1.cpp dene1.cpp dene2.cpp prg1k.cpp capma.cpp degiskenboyuts.cpp degiskenboyut.cpp
//her bir degiskenin sınırlarını görelim
#include <stdio.h>;
#include <limits.h>;
#include <float.h>;
#include <conio.h>;

int main() {

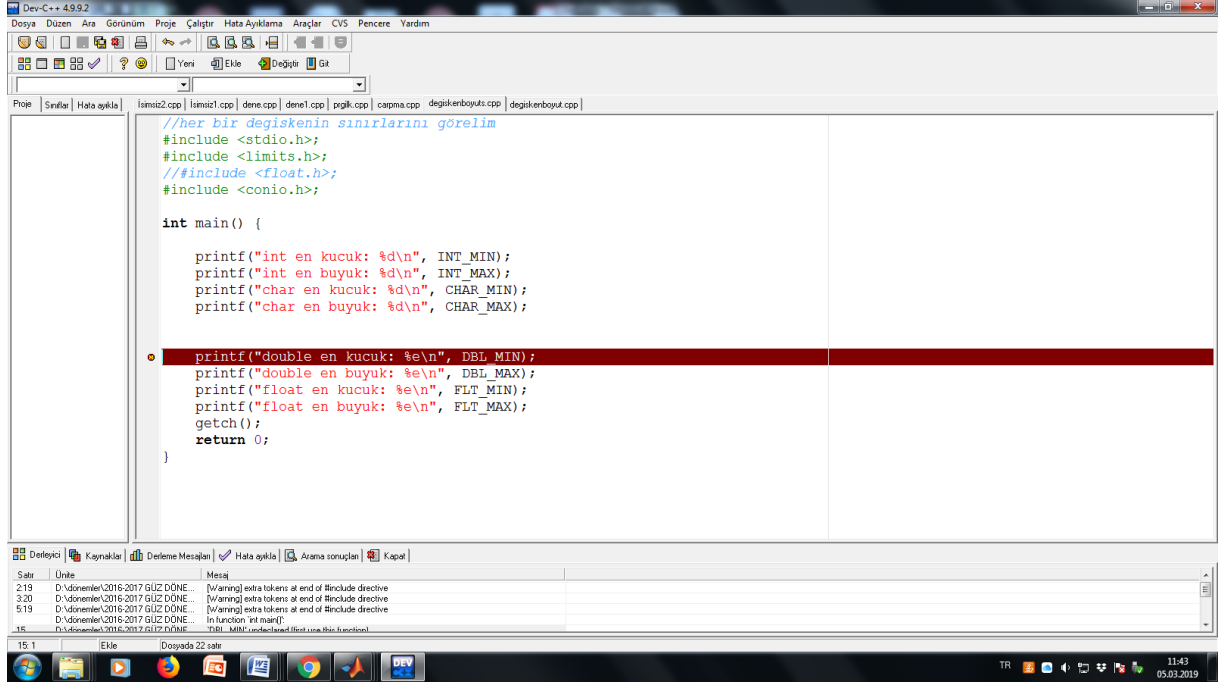
    printf("int en kucuk: %d\n", INT_MIN);
    printf("int en büyük: %d\n", INT_MAX);
    printf("char en kucuk: %d\n", CHAR_MIN);
    printf("char en büyük: %d\n", CHAR_MAX);

    printf("double en kucuk: %e\n", DBL_MIN);
    printf("double en büyük: %e\n", DBL_MAX);
    printf("float en kucuk: %e\n", FLT_MIN);
    printf("float en büyük: %e\n", FLT_MAX);
    getch();
    return 0;
}
D:\öğrenciler\2016-2017 GÜZ DÖNEMİ\Ist-205\degiskenboyuts.exe
int en kucuk: -2147483648
int en büyük: 2147483647
char en kucuk: -128
char en büyük: 127
double en kucuk: 2.225074e-308
double en büyük: 1.797693e+308
float en kucuk: 1.175494e-038
float en büyük: 3.402823e+038
```

Burada programa iki kütüphane daha eklendi, limits.h ve float.h.

limits.h → int ve char değişkenlerinin sınırlarının yazdırılması için, float.h → double ve float değişkenlerinin sınırlarını yazdırabilmek için eklendi. Bu kütüphaneleri tek tek silerek programı çalıştırarak nelerin değiştiğini gözlemleyebilirsiniz.

Örneğin, float.h kütüphanesini sildiğimizde programı çalıştırdığımızda aşağıdaki ekran ile karşılaşırız.



```
//her bir degiskenin sınırlarını görelim
#include <stdio.h>
#include <limits.h>
// #include <float.h>;
#include <conio.h>

int main() {

    printf("int en küçük: %d\n", INT_MIN);
    printf("int en büyük: %d\n", INT_MAX);
    printf("char en küçük: %d\n", CHAR_MIN);
    printf("char en büyük: %d\n", CHAR_MAX);

    printf("double en küçük: %e\n", DBL_MIN);
    printf("double en büyük: %e\n", DBL_MAX);
    printf("float en küçük: %e\n", FLT_MIN);
    printf("float en büyük: %e\n", FLT_MAX);
    getch();
    return 0;
}
```

15 1 | Ekke | Dosya 22 satır

219 D:\donemler\2016-2017 GÜZ DÖNE... [Warning] extra tokens at end of #include directive  
320 D:\donemler\2016-2017 GÜZ DÖNE... [Warning] extra tokens at end of #include directive  
519 D:\donemler\2016-2017 GÜZ DÖNE... [Warning] extra tokens at end of #include directive  
15 D:\donemler\2016-2017 GÜZ DÖNE... In function 'int main():  
D:\donemler\2016-2017 GÜZ DÖNE... DBL\_MIN undeclared (first use this function).

TR 11:43 05.03.2019