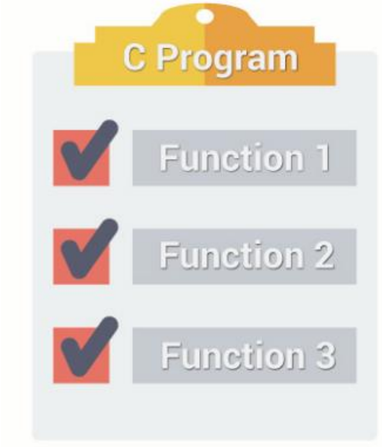
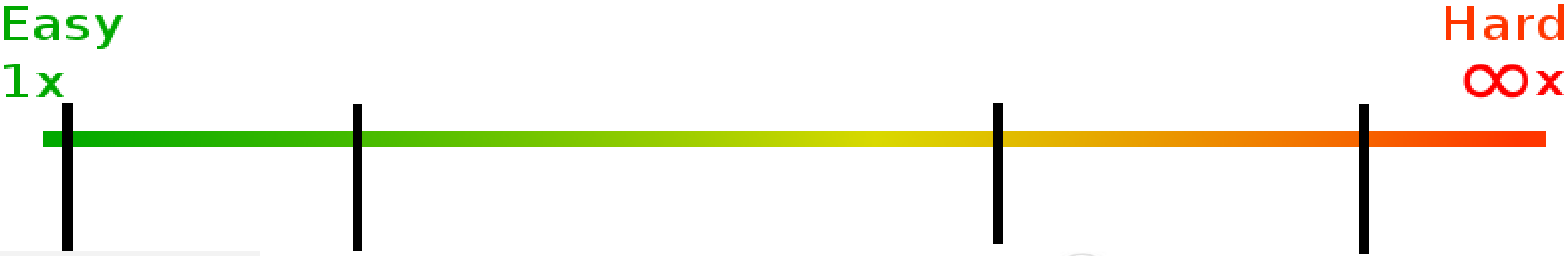
Fonksiyonlar



Onur GÖK

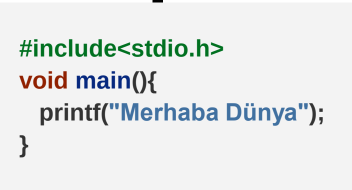
Fidan Kaya Gülagız

Fonksiyonlar

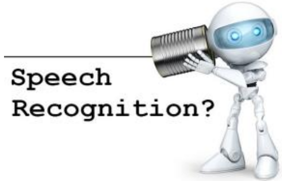


Easy
1x

Hard
∞x

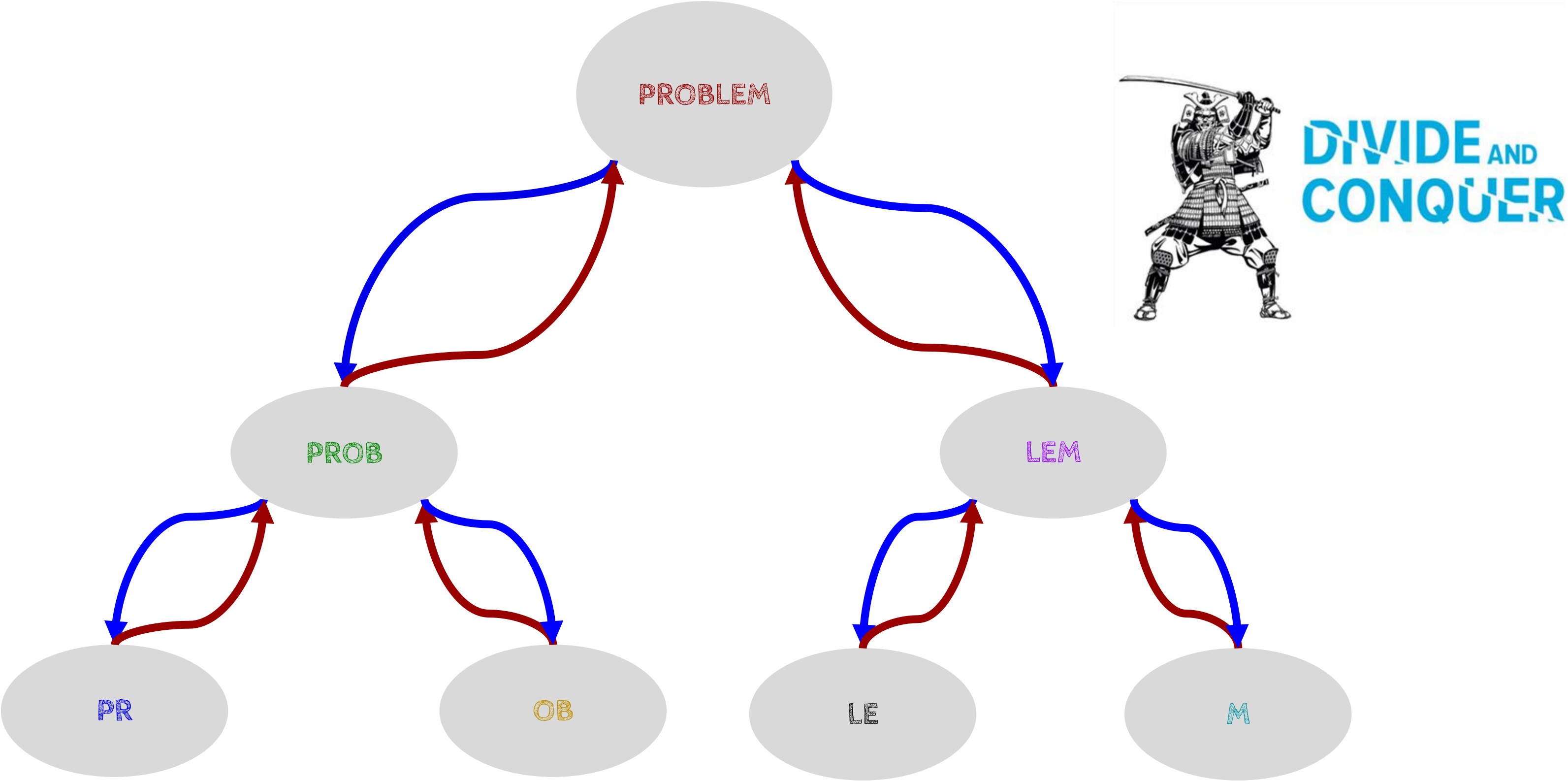


```
#include<stdio.h>
void main(){
  printf("Merhaba Dünya");
}
```



Speech
Recognition?

Fonksiyonlar



PROBLEM

PROB

LEM

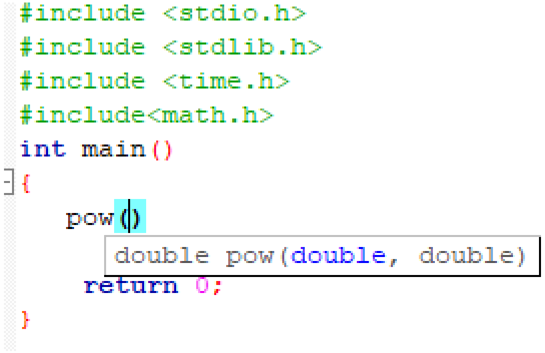
PR

OB

LE

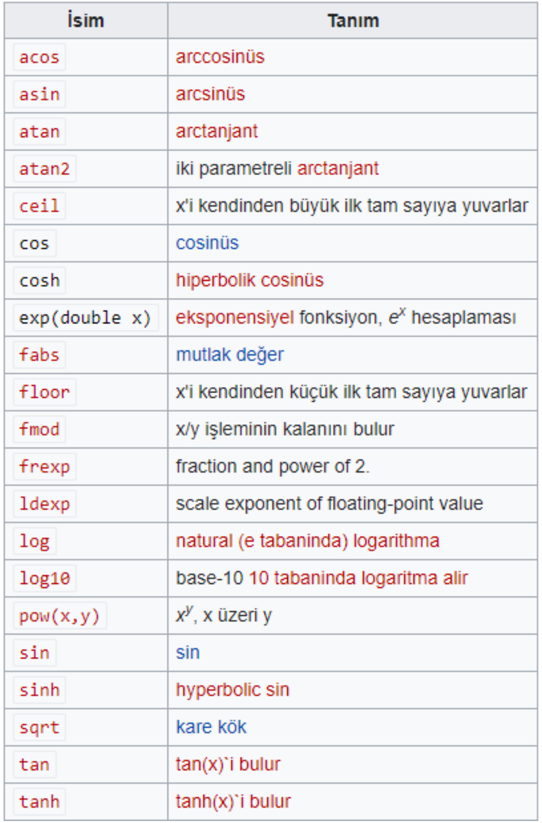
M

Hazır Fonksiyonlar



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include<math.h>
int main()
{
    pow()
    return 0;
}
```

Matematiksel Hazır Fonksiyonlar



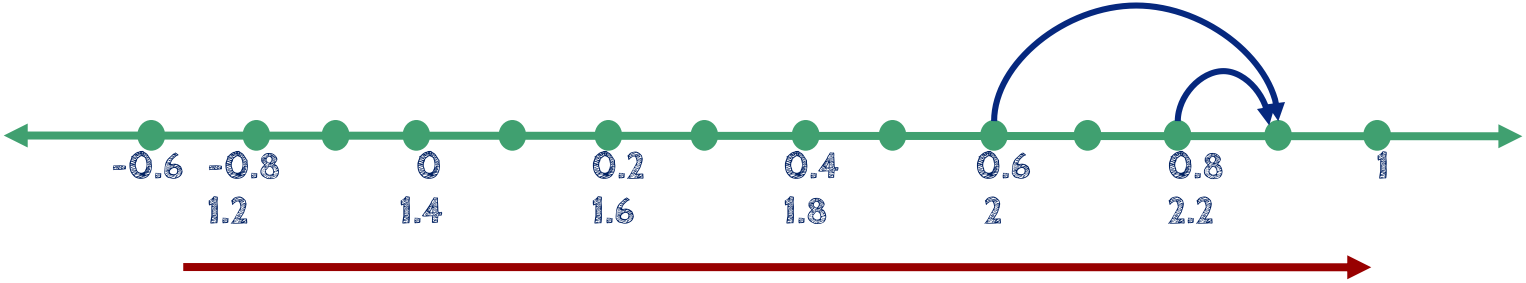
İsim	Tanım
acos	arccosinüs
asin	arcsinüs
atan	arctanjant
atan2	iki parametreli arctanjant
ceil	x'i kendinden büyük ilk tam sayıya yuvarlar
cos	cosinüs
cosh	hiperbolik cosinüs
exp(double x)	eksponensiyel fonksiyon, e^x hesaplaması
fabs	mutlak değer
floor	x'i kendinden küçük ilk tam sayıya yuvarlar
fmod	x/y işleminin kalanını bulur
frexp	fraction and power of 2.
ldexp	scale exponent of floating-point value
log	natural (e tabaninda) logarithma
log10	base-10 10 tabaninda logaritma alir
pow(x,y)	x^y, x üzeri y
sin	sin
sinh	hyperbolic sin
sqrt	kare kök
tan	tan(x)`i bulur
tanh	tanh(x)`i bulur

Kaynakça: https://tr.wikipedia.org/wiki/Math.h

Karekök & Üs

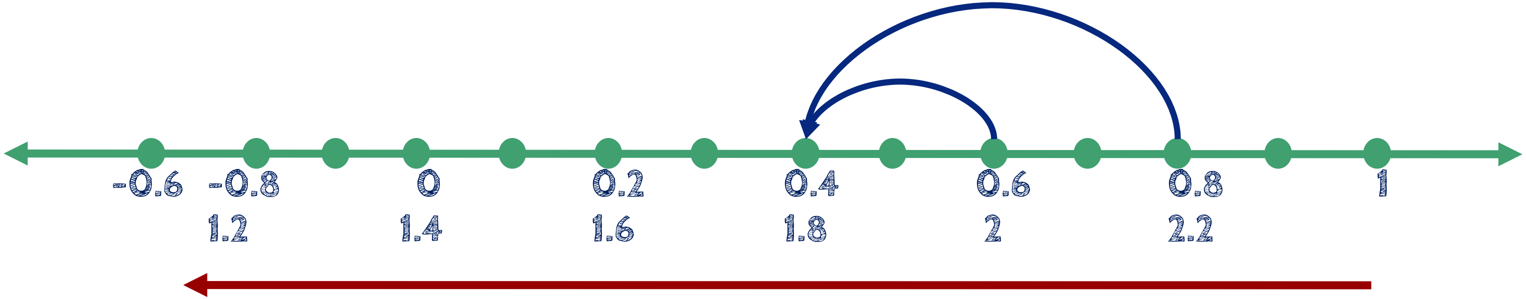
```
#include <stdio.h>
#include <math.h>
int main() {
        float sayi;
        float sonuc;
        printf("Sayi Giriniz : ");
        scanf("%f",&sayi);
        sonuc= sqrt(sayi);
        printf("sqrt(%f) : %f\n",sayi,sonuc);
        sonuc=pow(sayi,2);
        printf("pow(%f,2) : %f\n",sayi,sonuc);
        sonuc=pow(2,sayi);
        printf("pow(2,%f) : %f\n",sayi,sonuc);
        return 0;
}
```

ceil



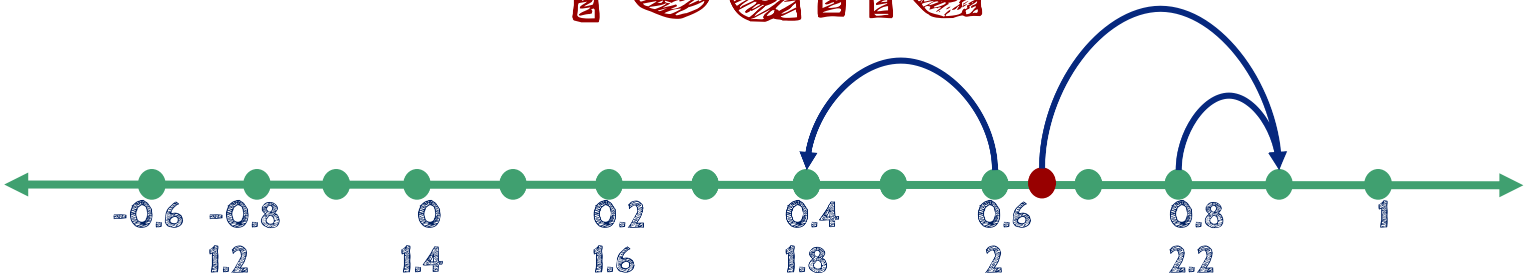
```
printf("ceil(%f) : %f\n",1.4,ceil(1.4));
printf("ceil(%f) : %f\n",1.8,ceil(1.8));
```

floor



```
printf("floor(%f) : %f\n",1.4,floor(1.4));
printf("floor(%f) : %f\n",1.8,floor(1.8));
```

round



```
printf("round(%f) : %f\n",1.4,round(1.4));
printf("round(%f) : %f\n",1.5,round(1.5));
printf("round(%f) : %f\n",1.8,round(1.8));
```

abs

```
printf("fabs(%f) : %f\n",-1.3,fabs(-1.3));
printf("abs(%d) : %f\n",-1,abs(-1));
```

Fonksiyonlar

C programlama dilinde fonksiyon tanımlanması için belirli kurallar vardır:

```
Tip FonksiyonAdı (Fonksiyon Giris Parametreleri)
{
  Deyimler...
  return fonksiyon tipinde deger
}
```

- Fonksiyon tipi
- Fonksiyon adı tanımlaması
- Fonksiyon giris parametreleri alanı
- return ifadesi

Fonksiyonlar

```
#include<stdio.h>
int main(        )
{




          return 0;
}
```

```
    /\
   /  \
  /    \
 /      \
----------
|        |
|        |
|        |
----------
```

Fonksiyonlar

```
#include<stdio.h>
int main(        )
{
        printf( "    /\\   \n" );
        printf( "   /  \\  \n" );
        printf( "  /    \\ \n" );
        printf( " /      \\ \n" );
        printf( "----------\n" );
        printf( "|        |\n" );
        printf( "|        |\n" );
        printf( "|        |\n" );
        printf( "----------\n" );

        return 0;
}
```

```
   /\
  /  \
 /    \
/      \
----------
|        |
|        |
|        |
----------
```

Fonksiyonlar

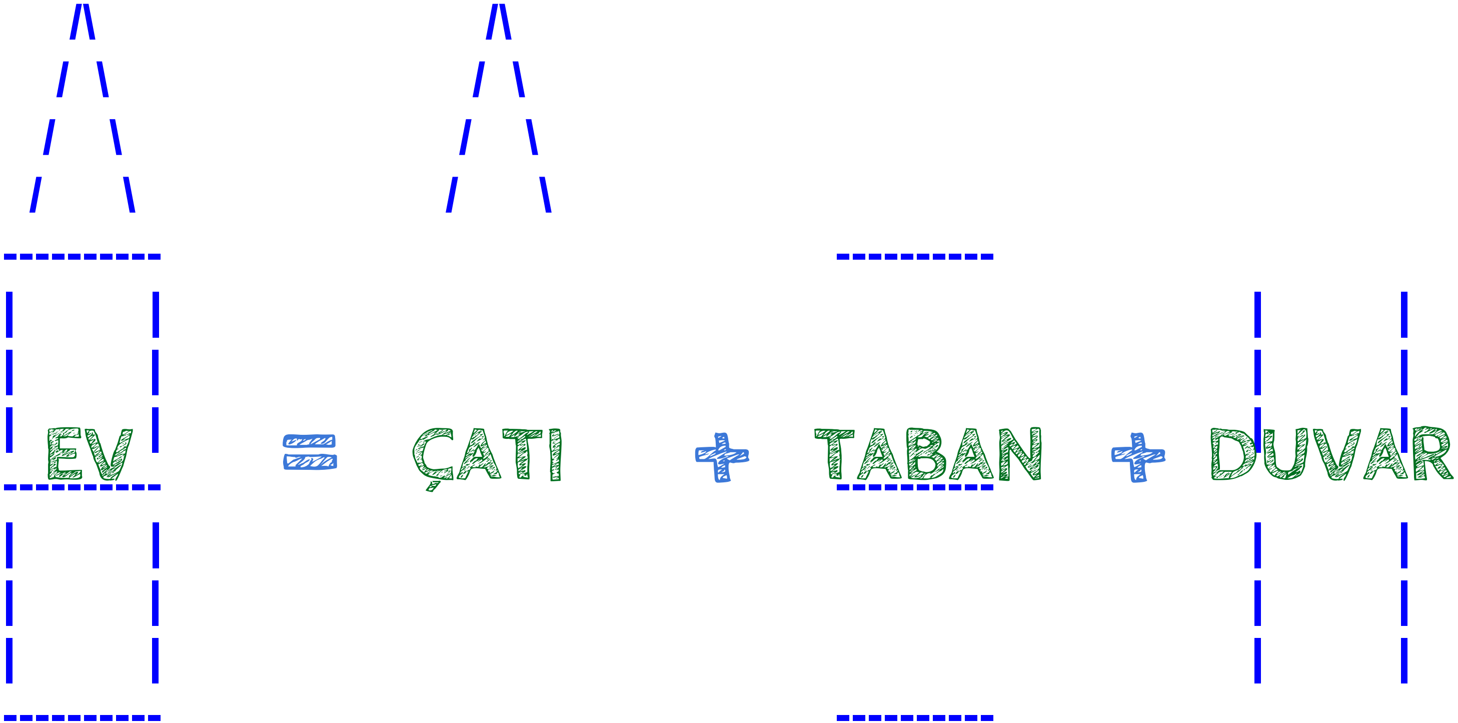
```
#include<stdio.h>
int main(        )
{
            printf( "    /\\   \n" );
            prin    "   /  \\  \n" );
            pr      /          );
            print              \\ \n" );
            print              ---\n" );
            p        |         n" );
               ntf( "|        |  " );
            printf( "|        |\n" );
            printf( "----------\n" );

            return 0;
}
```

```
    /\
   /  \
  /    \
 /      \
----------
|        |
|        |
|        |
----------
|        |
|        |
|        |
----------
```

Fonksiyonlar

DIVIDE AND CONQUER



EV = ÇATI + TABAN + DUVAR

Fonksiyonlar

```c
#include<stdio.h>
void catiyi_ciz( )
{
    printf( "    /\\    \n" );
    printf( "   /  \\   \n" );
    printf( "  /    \\  \n" );
    printf( " /      \\ \n" );
}
void duvar_ciz( )
{
    printf( "|        |\n" );
    printf( "|        |\n" );
    printf( "|        |\n" );
}
void taban_ciz( )
{
    printf( "----------\n" );
}
```

```c
int main( )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
return 0;
}
```

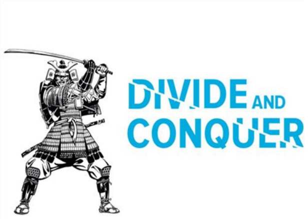
```
    /\
   /  \
  /    \
 /      \
----------
|        |
|        |
|        |
----------
|        |
|        |
|        |
----------
```

Fonksiyonlar

```
#include<stdio.h>
void catiyi_ciz( )
{
    printf( "    /\\    \n" );
    printf( "   /  \\   \n" );
    printf( "  /    \\  \n" );
    printf( " /      \\ \n" );
}
void duvar_ciz( )
{
    printf( "|        |\n" );
    printf( "|        |\n" );
    printf( "|        |\n" );
}
void taban_ciz( )
{
    printf( "----------\n" );
}
int main( )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );


return 0;
}
```

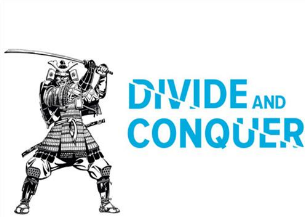
```
    /\
   /  \
  /    \
 /      \
----------
|        |
|        |
|        |
----------
|        |
|        |
|        |
----------
```



```
#include<stdio.h>
int main( )
{
                printf( "    /\\   \n" );
                printf( "   /  \\  \n" );
                printf( "  /    \\ \n" );
                printf( " /      \\ \n"
);
                printf( "----------\n"
);
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "----------\n"
);
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "----------\n"
);


                return 0;
}
```

Fonksiyonlar



```c
#include<stdio.h>
void catiyi_ciz( )
{
   printf( "    /\\    \n" );
   printf( "   /  \\   \n" );
   printf( "  /    \\  \n" );
   printf( " /      \\ \n" );
}
void duvar_ciz( )
{
   printf( "|        |\n" );
   printf( "|        |\n" );
   printf( "|        |\n" );
}
void taban_ciz( )
{
   printf( "----------\n" );
}
int main( )
{
   catiyi_ciz( );
   taban_ciz( );
   duvar_ciz( );
   taban_ciz( );
   duvar_ciz( );
   taban_ciz( );
   duvar_ciz( );
   taban_ciz( );
return 0;
}
```

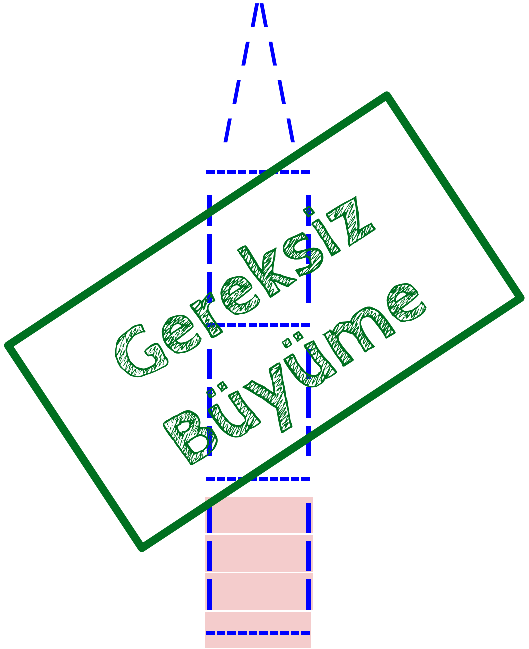
```
   /\
  /  \
 /    \
/      \
----------
|        |
|        |
|        |
----------
|        |
|        |
|        |
----------
|        |
|        |
|        |
----------
```

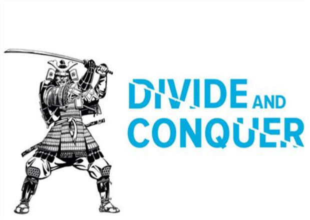
```c
#include<stdio.h>
int main( )
{
                    printf( "    /\\   \n" );
                    printf( "   /  \\  \n" );
                    printf( "  /    \\ \n" );
                    printf( " /      \\ \n"
);
                    printf( "----------\n"
);
                    printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n"
);
                    printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n"
);
                    printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n"
);

                    return 0;
}
```

Fonksiyonlar

```c
#include<stdio.h>
void catiyi_ciz( )
{
    printf( "    /\\    \n" );
    printf( "   /  \\   \n" );
    printf( "  /    \\  \n" );
    printf( " /      \\ \n" );
}
void duvar_ciz( )
{
    printf( "|        |\n" );
    printf( "|        |\n" );
    printf( "|        |\n" );
}
void taban_ciz( )
{
    printf( "----------\n" );
}
int main( )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
return 0;
}
```





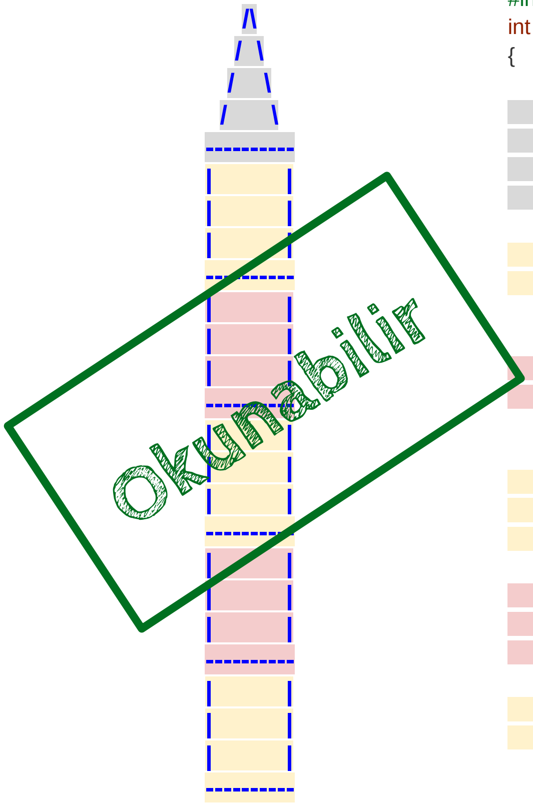
```c
#include<stdio.h>
int main( )
{
                              printf( "    /\\    \n" );
                              printf( "   /  \\   \n" );
                              printf( "  /    \\ \n" );
                              printf( " /      \\ \n"
);
                              printf( "----------\n"
);
                              printf( "|        |\n" );
                              printf( "|        |\n" );
                              printf( "|        |\n" );
                              printf( "----------\n"
);
                              printf( "|        |\n" );
                              printf( "|        |\n" );
                              printf( "|        |\n" );
                              printf( "----------\n"
);
                              printf( "|        |\n" );
                              printf( "|        |\n" );
                              printf( "|        |\n" );
                              printf( "----------\n"
);

                              return 0;
}
```

Fonksiyonlar

DIVIDE AND CONQUER

```
#include<stdio.h>
void catiyi_ciz( void )
{
    printf( "    /\\    \n" );
    printf( "   /  \\   \n" );
    printf( "  /    \\  \n" );
    printf( " /      \\ \n" );
}
void duvar_ciz( void )
{
    printf( "|        |\n" );
    printf( "|        |\n" );
    printf( "|        |\n" );
}
void taban_ciz( void )
{
    printf( "----------\n" );
}
int main( void )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
return 0;
}
```



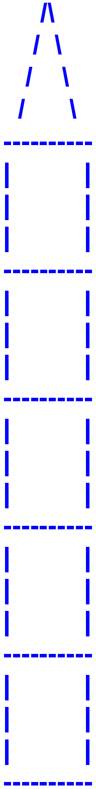
```
#include<stdio.h>
int main( void )
{
                                printf( "    /\\   \n" );
                                printf( "   /  \\  \n" );
                                printf( "  /    \\ \n" );
                                printf( " /      \\ \n" );
                                printf( "----------\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "----------\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "----------\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "----------\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "----------\n" );
                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "|        |\n" );
                                printf( "----------\n" );
                                return 0;
}
```

Fonksiyonlar

```
#include<stdio.h>
void catiyi_ciz( void )
{
    printf( "   /\\   \n" );
    printf( "  / \\  \n" );
    printf( " /   \\ \n" );
    printf( "/     \\ \n" );
}
void duvar_ciz( void )
{
    printf( "|      |\n" );
    printf( "|      |\n" );
    printf( "|      |\n" );
}
void taban_ciz( void )
{
    printf( "----------\n" );
}
int main( void )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
return 0;
}
```

taban çizerim

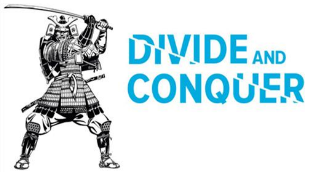
fonksiyon çagırırım



her is yapılır

```
#include<stdio.h>
int main( void )
{
        printf( "   /\\   \n" );
        printf( "  / \\  \n" );
        printf( " /   \\ \n" );
        printf( "/     \\ \n" );
        printf( "----------\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "----------\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "----------\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "----------\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "----------\n" );
    printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "|      |\n" );
        printf( "----------\n" );
        return 0;

}
```

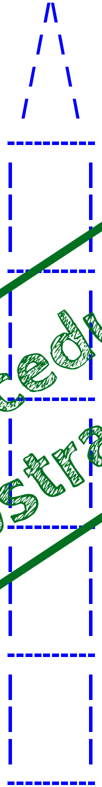


Fonksiyonlar

```
#include<stdio.h>
void catiyi_ciz( void )
{
    printf( "    /\\    \n" );
    printf( "   /  \\   \n" );
    printf( "  /    \\  \n" );
    printf( " /      \\ \n" );
}
void duvar_ciz( void )
{
    printf( "|        |\n" );
    printf( "|        |\n" );
    printf( "|        |\n" );
}
void taban_ciz( void )
{
    printf( "----------\n" );
}
int main( void )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
return 0;
}
```

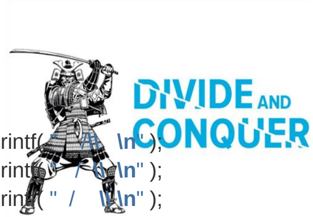
taban çizerim

fonksiyon çagırırım



procedural abstraction

her is yapılır

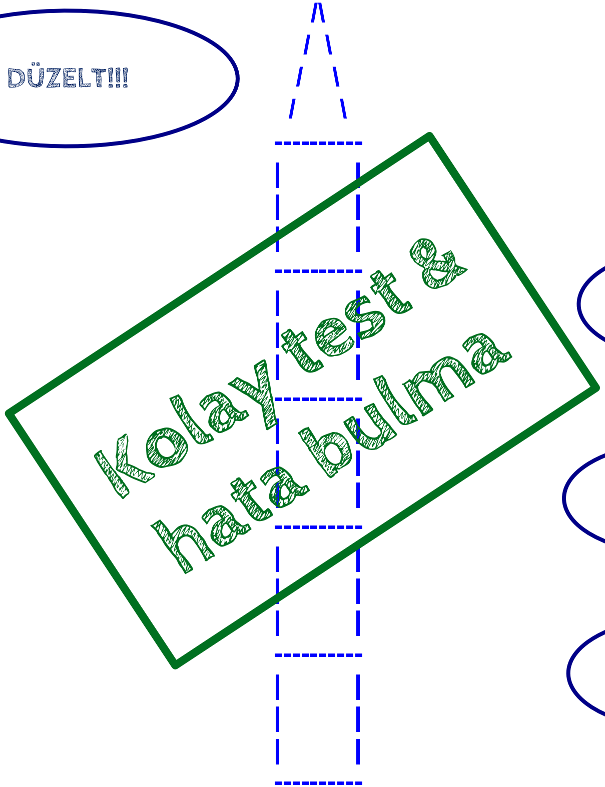


```
#include<stdio.h>
int main( void )
{
                printf( "    /\\    \n" );
                printf( "   /  \\   \n" );
                printf( "  /    \\  \n" );
                printf( " /      \\ \n"
);
                printf( "----------\n"
);
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "----------\n"
);
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "----------\n"
);
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "----------\n"
);
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "----------\n"
);
           printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "|        |\n" );
                printf( "----------\n"
);
```

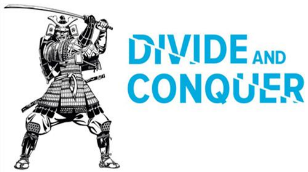
Fonksiyonlar

```
#include<stdio.h>
void catiyi_ciz( void )
{
    printf( "   /\\    \n" );
    printf( "  /  \\   \n" );
    printf( " /    \\  \n" );
    printf( "/      \\ \n" );
}
void duvar_ciz( void )
{
    printf( "|        |\n" );
    pri( "|        |\n" );
    printf( "|        |\n" );
}
void taban_ciz( void )
{
    printf( "----------\n" );
}
int main( void )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
return 0;
}
```

DÜZELT!!!



Kolay test & hata bulma



DIVIDE AND CONQUER

```
#include<stdio.h>
int main( void )
{
          printf( "   /\\   \n" );
          printf( "  / \\  \n" );
          printf( " /   \\ \n" );
          printf( "/     \\ \n" );
          printf( "----------\n" );
          printf( "|        |\n" );
          pri( "|        |\n" );
          printf( "|        |\n" );
          printf( "----------\n" );
          printf( "|        |\n" );
          pri( "|        |\n" );
          printf( "|        |\n" );
          printf( "----------\n" );
          printf( "|        |\n" );
          pri( "|        |\n" );
          printf( "|        |\n" );
          printf( "----------\n" );
          printf( "|        |\n" );
          pri( "|        |\n" );
          printf( "|        |\n" );
          printf( "----------\n" );
    printf( "|        |\n" );
          pri( "|        |\n" );
          printf( "|        |\n" );
          printf( "----------\n" );
          return 0;
}
```

DÜZELT!!
!

DÜZELT!!
!

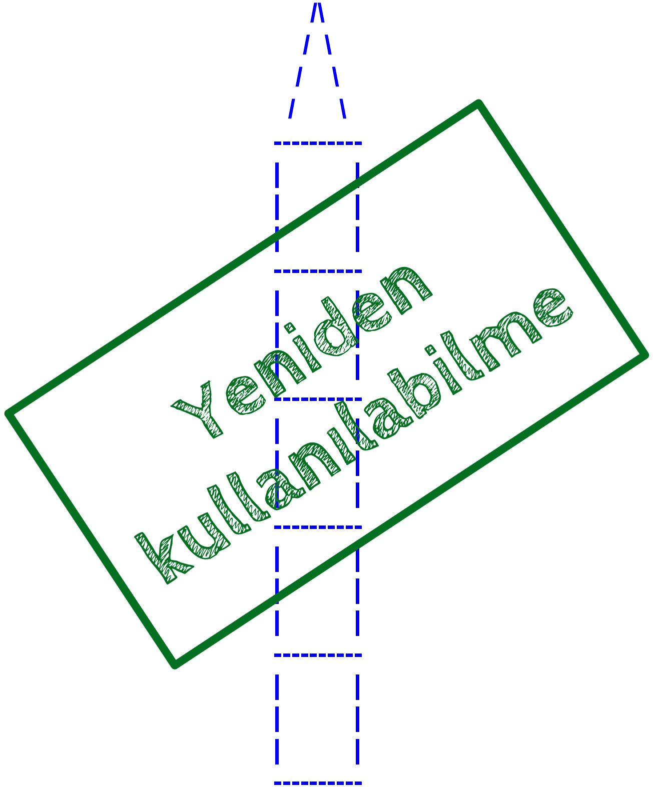
DÜZELT!!
!

DÜZELT!!
!

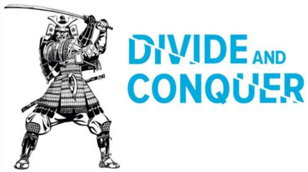
DÜZELT!!
!

Fonksiyonlar

```
#include<stdio.h>
void catiyi_ciz( void )
{
    printf( "    /\\    \n" );
    printf( "   /  \\   \n" );
    printf( "  /    \\ \n" );
    printf( " /      \\ \n" );
}
void duvar_ciz( void )                1
{
    printf( "|        |\n" );
    printf( "|        |\n" );
    printf( "|        |\n" );
}
void taban_ciz( void )
{
    printf( "----------\n" );
}
int main( void )
{
    catiyi_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
    duvar_ciz( );
    taban_ciz( );
return 0;
}
```

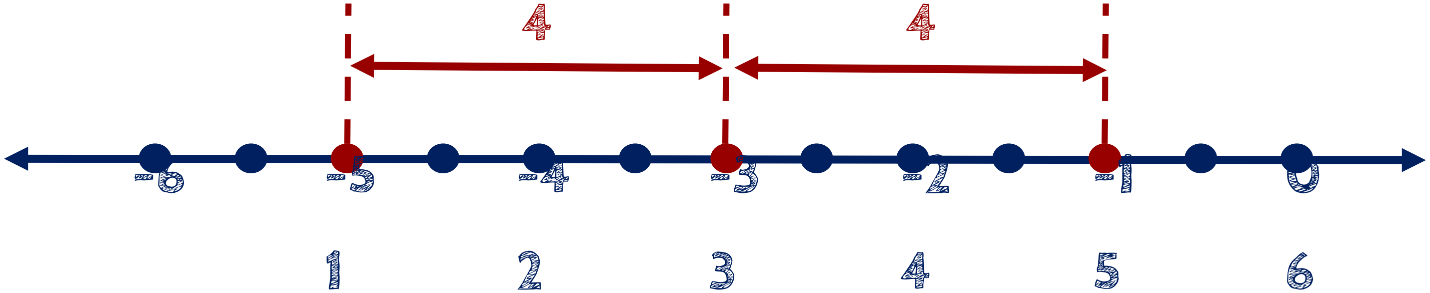


Yeniden kullanılabilme



```
#include<stdio.h>
int main( void )
{
                    printf( "    /\\   \n" );
                    printf( "   /  \\  \n" );
                    printf( "  /    \\ \n" );
                    printf( " /      \\ \n" );
                    printf( "----------\n" );
                    printf( "|        |\n" );
1                   printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n" );
                    printf( "|        |\n" );
2                   printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n" );
                    printf( "|        |\n" );
3                   printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n" );
                    printf( "|        |\n" );
4                   printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n" );
           printf( "|        |\n" );
5                   printf( "|        |\n" );
                    printf( "|        |\n" );
                    printf( "----------\n" );
                    return 0;
}
```

Mutlak Deger

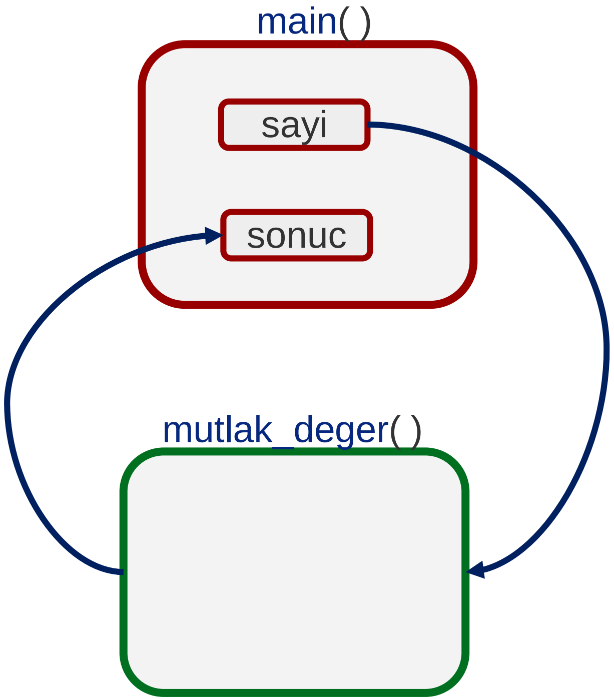


$$|-4| = |4|$$

Mutlak Deger

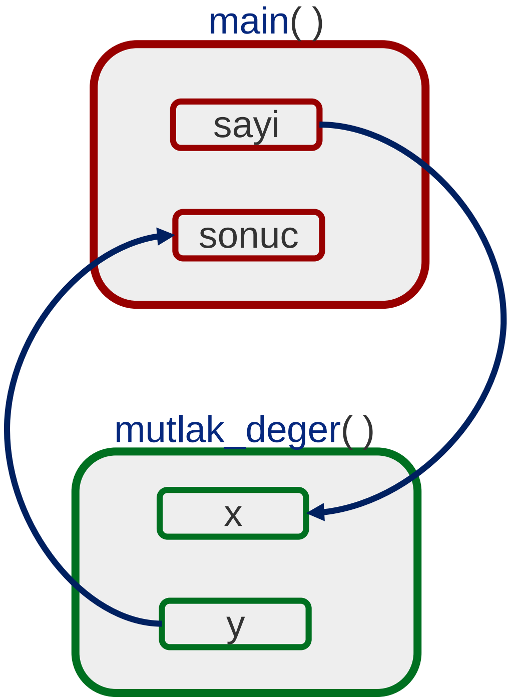
```
#include <stdio.h>
#include <stdlib.h>

int main( ) {
            int sayi, sonuc;
            printf("sayi girin:");
            scanf("%d", &sayi);
            sonuc = mutlak_deger(sayi);
            printf("mutlak deger: %d\n",
sonuc);
            return 0;
}
```



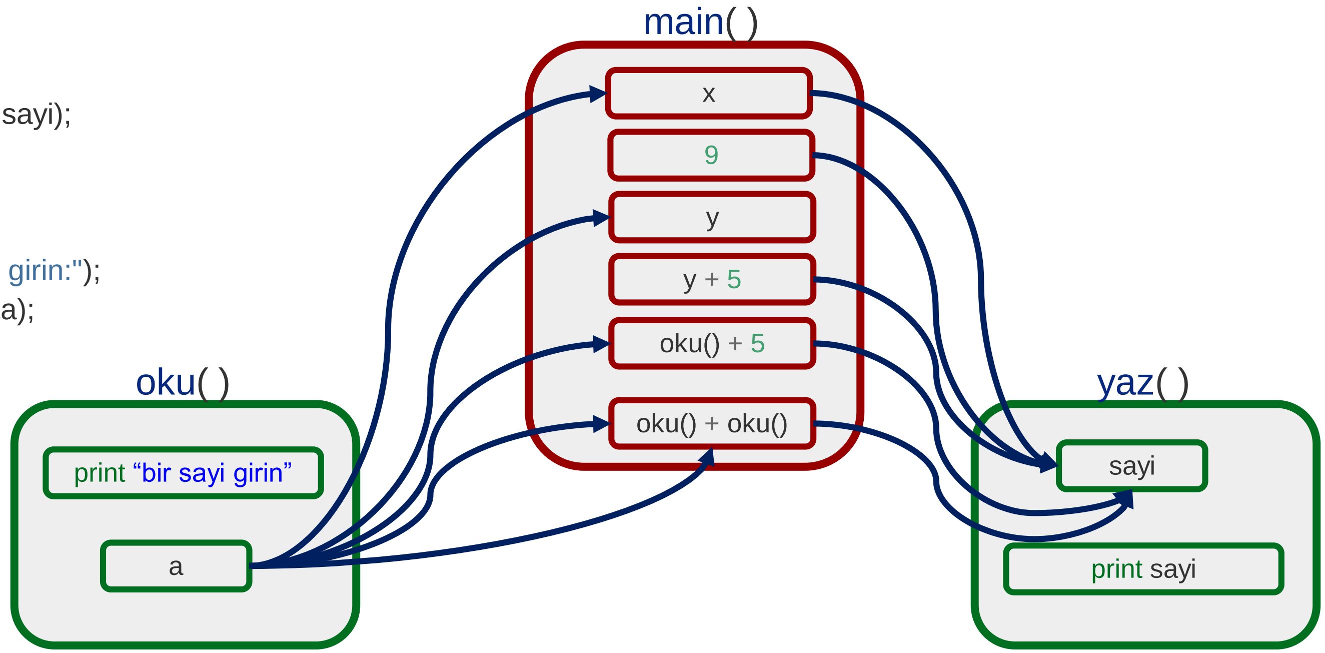
Mutlak Deger

```
int mutlak_deger( int x ) {
            int y;
            if (x >= 0)
            y = x;
            else
            y = -1 * x;
            return y;
}
```



printf & scanf

```
#include <stdio.h>
#include <stdlib.h>
void yaz(int sayi) {
            printf("%d\n", sayi);
}
int oku() {
            int a;
            printf("bir sayi girin:");
            scanf("%d", &a);
            return a;
}
int main() {
            int x, y;
            x = oku();
            yaz(x);
            yaz(9);
            y = oku();
            yaz(y + 5);
            yaz( oku() + 5);
            yaz( oku() + oku() );
            return 0;
}
```



Ucgen

analiz:

~~1. s 1 t~~

~~2. sa x~~

~~3. satirda ne x~~

~~4. satirda tane x~~

~~5. satirda 2 tane x~~

~~6. sati ane x~~

~~7. s ne x~~

~~8. da e x~~

~~9. satirda 4 tane x~~

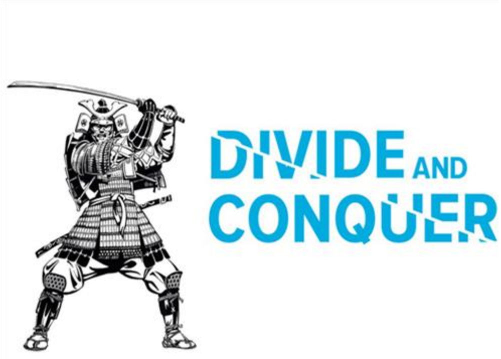
x
xx
xxx
x
xx
x
xx
xxx
xxxx

Ucgen

analiz:

1. satirda	1 tane x
2. satirda	2 tane x
3. satirda	3 tane x
1. satirda	1 tane x
2. satirda	2 tane x
1. satirda	1 tane x
2. satirda	2 tane x
3. satirda	3 tane x
4. satirda	4 tane x

```
x
xx
xxx
x
xx
x
xx
xxx
xxxx
```



Hatırlatma

x

x x

x x x

x x x x

x x x x x

Hatırlatma

```
#include <stdio.h>
int main() {
        int i, j;
        for (i = 1 ; i <= 5 ; i++) {
                for (j = 1 ; j <= i
; j++) {

        printf("x");
                }
                printf("\n");
        }
        return 0;
}
```

satir =5

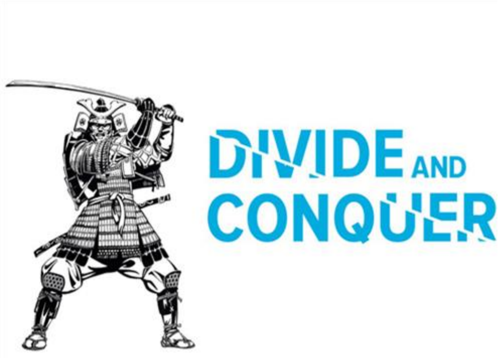
sutun = 5

Ucgen

```c
#include <stdio.h>
#include <stdlib.h>
void ucgen_ciz(int x) {
        int i, j;
        for (i = 1 ; i <= x ; i++) {
                for (j = 1 ; j <= i ; j++) {
                        printf("*");
                }
                printf("\n");
        }
}

int main() {
        ucgen_ciz(3);
        ucgen_ciz(2);
        ucgen_ciz(4);
        return 0;
}
```

```
x
xx
xxx
x
xx
x
xx
xxx
xxxx
```



Ucgen

analiz:

1. satirda	1 tane x
2. satirda	2 tane x
3. satirda	3 tane x
4. satirda	4 tane x
1. satirda	3 tane x
2. satirda	2 tane x
3. satirda	1 tane x

x

xx

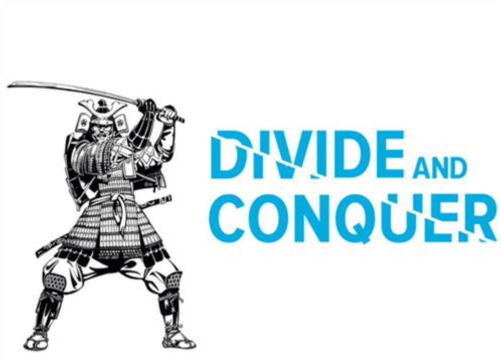
xxx

xxxx

xxx

xx

x



Ucgen

```c
#include <stdio.h>
#include <stdlib.h>
void ucgen_ciz(int x) {
        int i, j;
        for (i = 1 ; i <= x ; i++) {
                for (j = 1 ; j <= i ; j++) {
                        printf("*");
                }
                printf("\n");
        }
}
void ters_ucgen_ciz(int x) {
        int i, j;
        for (i = 1 ; i <= x ; i++) {
                for (j = 1 ; j <= x-i+1 ; j++) {
                        printf("*");
                }
                printf("\n");
        }
}
int main() {
        ucgen_ciz(4);
        ters_ucgen_ciz(3);
        return 0;
}
```

```
X
XX
XXX
XXXX
XXX
XX
X
```

Double Equality

```
#include <stdio.h>
#include <stdlib.h>
int main() {
            double x = 1.2;
            double y = 5.3;
            double z = y - 4.1;
            if ( z == x)
                        printf("esit\n");
            else
                        printf("esit degil\n");
            return 0;
}
```

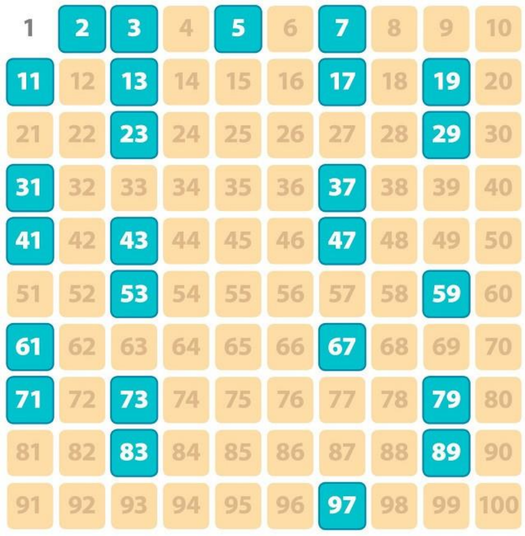
Double Equality

```c
#include <stdio.h>
#include <stdlib.h>
int esit_mi(double A, double B, double hata_payi) {
        if (fabs(A - B) < hata_payi) {
                return 1;
        } else {
                return 0;
        }
}
int main() {
        double x = 1.2;
        double y = 5.3;
        double z = y - 4.1;
        if ( esit_mi(x, z, 0.0001) == 1)
                printf("esit\n");
        else
                printf("esit degil\n");
        return 0;
}
```

HATA PAYI

0.0001

Asal Sayı



1	**2**	**3**	4	**5**	6	**7**	8	9	10
11	12	**13**	14	15	16	**17**	18	**19**	20
21	22	**23**	24	25	26	27	28	**29**	30
31	32	33	34	35	36	**37**	38	39	40
41	42	**43**	44	45	46	**47**	48	49	50
51	52	**53**	54	55	56	57	58	**59**	60
61	62	63	64	65	66	**67**	68	69	70
71	72	**73**	74	75	76	77	78	**79**	80
81	82	**83**	84	85	86	87	88	**89**	90
91	92	93	94	95	96	**97**	98	99	100

23 asal mıdır?

23 <= 4 X 4

2 <= 23/2

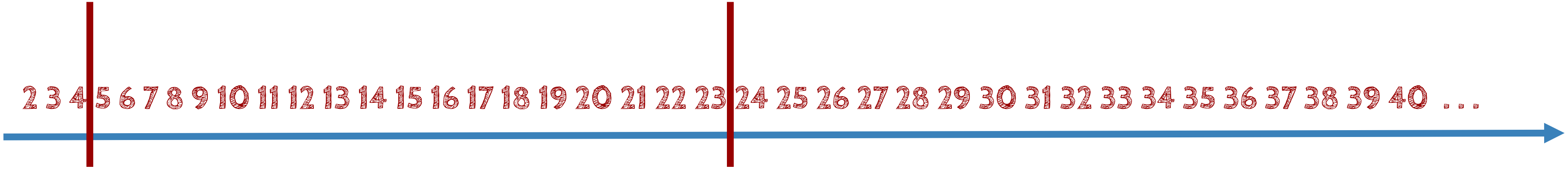
3 <= 23/3

4 <= 23/4

~~5 <= 23/5~~

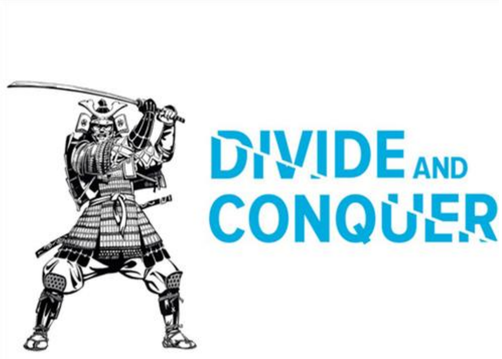
23 >= P X P

23 >= 4 X 4



Asal Sayı

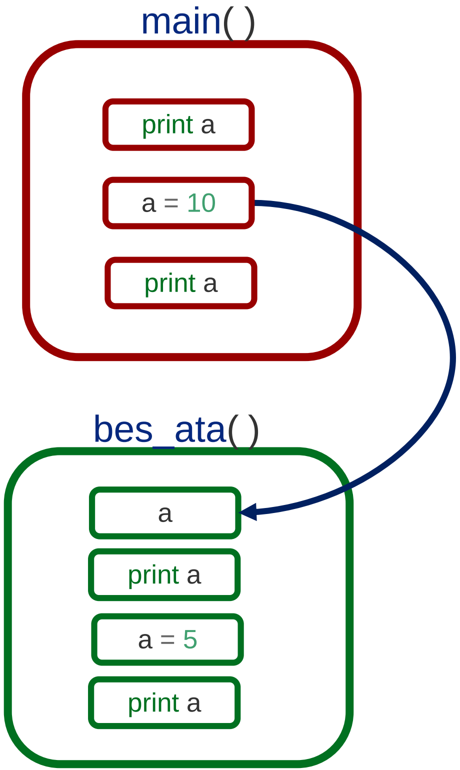
```c
#include <stdio.h>
#include <stdlib.h>
int asal_mi(int sayi) {
    int j,i=sayi;
        for(j = 2; j <= (i/j); j++)
            if(!(i%j)) break;
        if(j > (i/j)) return 1;
    return 0;
}
int main() {
    int a, b;
    printf("iki sayi girin: ");
    scanf("%d %d", &a, &b);
    int sayi;
    for (sayi = a ; sayi <= b ; sayi++) {
        int sonuc = asal_mi(sayi);
        if (sonuc == 1)           printf("asal: %d\n", sayi);
    }
    return 0;
}
```



main & fonksiyon

```c
#include <stdio.h>
void bes_ata(int a) {
    printf("alınan -> %d\n", a);
    a = 5;
    printf("degistirildi -> %d\n\n\n", a);
}

int main() {
    int a = 10;
    printf("gonderilmeden once -> %d\n\n\n", a);
    bes_ata(a);
    printf("gonderildikten sonra -> %d\n\n\n", a);
    return 0;
}
```



Sorular