

ÖĞRENCİ İSMİ:

SÜRENİZ 10 DAKİKADIR.

DERS NO(3 basamaklı):

BİL141 KISA SINAV VI ŞUBE 1 GRUP B SORULARI

- ASİSTANLARIN SINAV ESNASINDAKİ TEK GÖREVİ SINIFI GÖZETLEMektir. SORULARDAN PUAN ALABİLMENİZ İÇİN **TAM OLARAK CEVAP VERMENİZ GEREKMEKTEDİR**. KISMI PUANLAMA YAPILMAMAKTADIR. Örneğin, cevap 14.3 ise 14.30 yazan öğrenci o sorudan puan alamaz! / ve \ farkı, ' ve " farkı, $\leq$ yerine $\leq$ gibi detaylar da dâhil olmak üzere **CEVABINIZIN TAM OLARAK DOĞRULUĞUNA** emin olunuz. Ancak ,(nokta) yerine ,(virgül) koymak ya da C yerine c yazmak gibi durumlar, soruda özellikle ölçülenler arasında değilse , puan kaybına yol açmaz.
- EKRAN ÇIKTISI SORULARINDA KODLARIN ÇIKTISINI **HER HÜCRE BİR KARAKTERİ TEMSİL EDECEK ŞEKİLDE** KUTULARA YAZINIZ. Cevabınızın kutuya sığmadığı durumlar olursa, ilgili kısma ek hücreler çizerek cevabınızı belirtebilirsiniz. Hata içerdiğini düşündüğünüz sorularda (mantık hatası haricindekilerde) ne tür bir hata olduğunu aşağıdaki seçeneklerden seçerek cevap kutusunun içine (örn. "HATA1" şeklinde) **yazınız VE nedenini** cevap kutusunun altına ya da soruya yakın başka boş bir alana (örn. "Çünkü koddaki ... eksik." şeklinde) belirtiniz. HATA1:Kod hiç derlenemeyecektir. HATA2:Kod derlenecek ama çalışması esnasında hata verecektir. HATA3:Kod derlenecek ama çalışması esnasında sonsuz döngüye girecektir. Yukarıdaki ifadelere uymayan hatalarda (örneğin mantık hatası içerdiğini düşündüğünüz sorularda) hatasız sorularda olduğu gibi sadece ekran çıktısını yazınız, HATA# vs. yazmayınız.
- KODDA BOŞLUK SORULARINDA EKRAN ÇIKTISINI SAĞLAYACAK ŞEKİLDE BOŞLUĞU KULLANDIĞINIZ **TÜM KARAKTERLERE** ($\leq$ yerine $\leq$ gibi) **DİKKAT EDEREK DOLDURUNUZ**. Gereksiz komut/ifadeler eklemeyiniz; bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

// kod.c dosyası

#include <stdio.h>

#include <string.h>

int main(int argc, char* argv[])

{

int y = (int)argv[1][0] - (int)'0' + 1;

char d[10];

strcpy(d, argv[2]);

strcat(d, argv[3]);

while(y>0)

{

y=y-2;

printf("%s\n", d);

}

return 0;

}

Komut Ekranı

gcc kod.c -o kod.exe
kod.exe 43 kere 41 kac eder?

k	e	r	e	4	1				
k	e	r	e	4	1				
k	e	r	e	4	1				

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

#include <stdio.h>

int t(int[], int);

int main(int argc, char* argv[])

{

int dizi[4] = {2,5,8};

printf("%d", t(dizi,argc));

return 0;

}

int t(int dizi[], int x)

{

if(x<=2) return dizi[x];

return t(dizi,x-1)+t(dizi,x-2)+t(dizi,x-3);

}

Komut Ekranı

gcc kod.c -o kod.exe
kod azman ve gargamel

2	8								

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

KODDA BOŞLUK DOLDURMA SORUSU

```
#include <stdio.h>
typedef struct
{
    double g;
} h;

int main()
{
    h hane[10]= {};
    int i;

    for(i=1; i<6; i++)
    {
        printf("%d.hane gelir:", i);
        scanf("%lf", &hane[ i ].g);

        hane[ 8 ].g += hane[ i ].g;
    }

    hane[ 8 ].g /= 5.0;
    for(i=1; i<6; i++)
    {
        if( hane[ i ].g > hane[ 8 ].g )
            printf("%d. hane ortalama ustü gelire sahip.\n", i);
    }
    return 0;
}
```

ÖRNEK EKRAN ÇIKTISI

```
1.hane gelir:100.0
2.hane gelir:120.0
3.hane gelir:140.0
4.hane gelir:180.0
5.hane gelir:155.0
3. hane ortalama ustü gelire sahip.
4. hane ortalama ustü gelire sahip.
5. hane ortalama ustü gelire sahip.

-----
Process exited after 33.02 seconds wi
Devam etmek için bir tuşa basın . . .
```

Kullanıcıdan 5 haneye(konuta) ait gelir değerini alıp, ortalama üstü değere sahip haneleri ekrana yazdıran kod. Kod çalıştırıldıktan sonra kullanıcı "100.0", "120.0", "140.0", "180.0" ve "155.0" değerlerini girmiş ve her girişten sonra enter'a basmıştır. Bu değerlerin ortalaması 139.0 ettiği için, 140.0, 155.0 ve 180.0 değerlerine sahip olan haneler "ortalama üstü" olarak ekrana yazdırılmıştır.

// kod.c dosyasi

```
#include <stdio.h>

void eb(char, int);

int main(int argc, char *argv[])
{
    printf("Y");

    int x= (int)argv[1][0] - (int)'0';

    eb('E',x);

    eb('R',argc);

    return 0;
}

void eb(char x, int y)
{
    int j;

    for(j=1; j<=y; j++)
        printf("%c",x);

    return;
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
kod.exe 3
```

Y	E	E	E	R	R				

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

BİL141 KISA SINAV VI ŞUBE 1 GRUP C SORULARI

- ASİSTANLARIN SINAV ESNASINDAKİ TEK GÖREVİ SINIFI GÖZETLEMektir. SORULARDAN PUAN ALABİLMEİNİZ İÇİN **TAM OLARAK CEVAP VERMENİZ GEREKMEKTEDİR**. KİSMİ PUANLAMA YAPILMAMAKTADIR. Örneğin, cevap 14.3 ise 14.30 yazan öğrenci o sorudan puan alamaz! / ve \ farkı, ' ve " farkı, ≤ yerine <= gibi detaylar da dâhil olmak üzere **CEVABINIZIN TAM OLARAK DOĞRULUĞUNA** emin olunuz. Ancak .(nokta) yerine ,(virgöl) koymak ya da C yerine c yazmak gibi durumlar, soruda özellikle ölçülenler arasında değilse , puan kaybına yol açmaz.
- EKRAN ÇIKTISI SORULARINDA KODLARIN ÇIKTISINI **HER HÜCRE BİR KARAKTERİ TEMSİL EDECEK ŞEKİLDE** KUTULARA YAZINIZ. Cevabınızın kutuya sığmadığı durumlar olursa, ilgili kısma ek hücreler çizerek cevabınızı belirtebilirsiniz. Hata içerdiğini düşündüğünüz sorularda (mantık hatası haricindekilerde) ne tür bir hata olduğunu aşağıdaki seçeneklerden seçerek cevap kutusunun içine (örn. "HATA1" şeklinde) **yazınız VE nedenini** cevap kutusunun altına ya da soruya yakın başka boş bir alana (örn. "Çünkü koddaki ... eksik." şeklinde) belirtiniz. HATA1:Kod hiç derlenemeyecektir. HATA2:Kod derlenecek ama çalışması esnasında hata verecektir. HATA3:Kod derlenecek ama çalışması esnasında sonsuz döngüye girecektir. Yukarıdaki ifadelere uymayan hatalarda (örneğin mantık hatası içerdiğini düşündüğünüz sorularda) hatasız sorularda olduğu gibi sadece ekran çıktısını yazınız, HATA# vs. yazmayınız.
- KODDA BOŞLUK SORULARINDA EKRAN ÇIKTISINI SAĞLAYACAK ŞEKİLDE BOŞLUĞU KULLANDIĞINIZ **TÜM KARAKTERLERE** (≤ yerine <= gibi) **DIKKAT EDEREK DOLDURUNUZ**. Gereksiz komut/ifadeler eklemeyiniz; bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

// kod.c dosyası

#include <stdio.h>

void eb(char, int);

int main(int argc, char *argv[])

```
{
    printf("K");
    int x= (int)argv[1][0] - (int)'0';
    eb('A',x);
    eb('R',argc);
    return 0;
}
```

void eb(char x, int y)

```
{
    int j;
    for(j=1; j<=y; j++)
        printf("%c",x);
    return;
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
kod.exe 3
```

K A A A R R

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

// kod.c dosyası

#include <stdio.h>

#include <string.h>

int main(int argc, char* argv[])

```
{
    int y= (int)argv[1][0] - (int)'0';
    char d[10];
    strcpy(d, argv[2]);
    strcat(d, argv[4]);
    while(y>0)
    {
        y=y-2;
        printf("%s\n", d);
    }
    return 0;
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
kod.exe 43 kere 41 kac eder?
```

k e r e k a c
k e r e k a c

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

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

```
int t(int[], int);
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    int dizi[4] = {3,6,9};
```

```
    printf("%d", t(dizi,argc));
```

```
    return 0;
```

```
}
```

```
int t(int dizi[], int x)
```

```
{
```

```
    if(x<=2) return dizi[x];
```

```
    return t(dizi,x-1)+t(dizi,x-2)+t(dizi,x-3);
```

```
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
kod azman ve gargamel
3 3
```

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

KODDA BOŞLUK DOLDURMA SORUSU

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

```
typedef struct
```

```
{
```

```
    double g;
```

```
} h;
```

```
int main()
```

```
{
```

```
    h hane[10]= {};
```

```
    int i;
```

```
    for(i=1; i<6; i++)
```

```
    {
```

```
        printf("%d.hane gelir:", i );
```

```
        scanf("%lf", &hane[ i ].g);
```

```
        hane[ 7 ].g += hane[ i ].g;
```

```
    }
```

```
    hane[ 7 ].g /= 5.0;
```

```
    for(i=1; i<6; i++)
```

```
    {
```

```
        if( hane[ i ].g > hane[ 7 ].g )
```

```
            printf("%d. hane ortalama ustü gelire sahip.\n",i);
```

```
    }
```

```
    return 0;
```

```
}
```

ÖRNEK EKRAN ÇIKTISI

```
1.hane gelir:100.0
2.hane gelir:120.0
3.hane gelir:140.0
4.hane gelir:180.0
5.hane gelir:155.0
3. hane ortalama ustü gelire sahip.
4. hane ortalama ustü gelire sahip.
5. hane ortalama ustü gelire sahip.
```

```
-----
Process exited after 33.02 seconds wi
Devam etmek için bir tuşa basın . . .
```

Kullanıcıdan 5 haneye(konuta) ait gelir değerini alıp, ortalama üstü değere sahip haneleri ekrana yazdıran kod. Kod çalıştırdıktan sonra kullanıcı "100.0", "120.0", "140.0", "180.0" ve "155.0" değerlerini girmiş ve her girişten sonra enter'a basmıştır. Bu değerlerin ortalaması 139.0 ettiği için, 140.0, 155.0 ve 180.0 değerlerine sahip olan haneler "ortalama üstü" olarak ekrana yazdırılmıştır.

ÖĞRENCİ İSMİ:

SÜRENİZ 10 DAKİKADIR.

DERS NO(3 basamaklı):

BİL141 KISA SINAV VI ŞUBE 2 GRUP D SORULARI

- ASİSTANLARIN SINAV ESNASINDAKİ TEK GÖREVİ SINIFI GÖZETLEMektir. SORULARDAN PUAN ALABİLMENİZ İÇİN TAM OLARAK CEVAP VERMENİZ GEREKMEKTEDİR. KİSMİ PUANLAMA YAPILMAMAKTADIR. Örneğin, cevap 14.3 ise 14.30 yazan öğrenci o sorudan puan alamaz! / ve \ farkı, ' ve " farkı, ≤ yerine <= gibi detaylar da dâhil olmak üzere CEVABINIZIN TAM OLARAK DOĞRULUĞUNA emin olunuz. Ancak .(nokta) yerine ,(virgül) koymak ya da C yerine c yazmak gibi durumlar, soruda özellikle ölçülenler arasında değilse , puan kaybına yol açmaz.
- EKRAN ÇIKTISI SORULARINDA KODLARIN ÇIKTISINI HER HÜCRE BİR KARAKTERİ TEMSİL EDECEK ŞEKİLDE KUTULARA YAZINIZ. Cevabınızın kutuya sığmadığı durumlar olursa, ilgili kısma ek hücreler çizerek cevabınızı belirtebilirsiniz. Hata içerdiğini düşündüğünüz sorularda (mantık hatası haricindekilerde) ne tür bir hata olduğunu aşağıdaki seçeneklerden seçerek cevap kutusunun içine (örn. "HATA1" şeklinde) yazınız VE nedenini cevap kutusunun altına ya da soruya yakın başka boş bir alana (örn. "Çünkü koddaki ... eksik." şeklinde) belirtiniz. HATA1:Kod hiç derlenemeyecektir. HATA2:Kod derlenecek ama çalışması esnasında hata verecektir. HATA3:Kod derlenecek ama çalışması esnasında sonsuz döngüye girecektir. Yukarıdaki ifadelere uymayan hatalarda (örneğin mantık hatası içerdiğini düşündüğünüz sorularda) hatasız sorularda olduğu gibi sadece ekran çıktısını yazınız, HATA# vs. yazmayınız.
- KODDA BOŞLUK SORULARINDA EKRAN ÇIKTISINI SAĞLAYACAK ŞEKİLDE BOŞLUĞU KULLANDIĞINIZ TÜM KARAKTERLERE (≤ yerine <= gibi) DİKKAT EDEREK DOLDURUNUZ. Gereksiz komut/ifadeler eklemeyiniz; bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

// kod.c dosyası

#include <stdio.h>

int f(int a);

int main(int argc, char* argv[])

{

printf("R");

printf("%d", f(argc));

return 0;

}

int f(int a)

{

if(a==1) return 3;

else

return f(a-2)*2;

}

Komut Ekranı

gcc kod.c -o kod.exe
kod.exe 141 dersi ve ogrettikleri

R 1 2

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

// kod.c dosyası

#include <stdio.h>

int main(int argc, char* argv[])

{

printf("%s", (argc%3==0) ? argv[2]:argv[1]);

// %sden sonra \n yok

printf("%d",argc);

return 0;

}

Komut Ekranı

gcc kod.c -o kod.exe
kod.exe temmuzda okulda

o k u l d a 3

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

KODDA BOŞLUK DOLDURMA SORUSU

```
#include <stdio.h>
void triangle(int, int);
void bos (int);
void spaceStar(int, int);
int main()
{
    int boyut;
    while(1)
    {
        printf("Boyut:");
        scanf("%d", &boyut);
        triangle(boyut,0);
        printf("\n\n");
    }
    return 0;
}
void triangle(int n, int h)
{
    if(n==1)
    {
        spaceStar(n,h);
        return;
    }
    int i;
    spaceStar(n,h);
    printf("\n");
    triangle( n-1 , h+1 );
    printf("\n");
}
```

bos(h)

```
for(i=1; i<=n; i++)
    printf(" "); //1 yıldız ve 1 bosluk
```

```
void bos (int x)
{
    int i;
    for(i=1; i<=x; i++)
        printf(" "); //2 bosluk
}
```

```
void spaceStar(int n, int bosluk)
{
    int i;
    bos(bosluk);
    for(i=1; i<=n; i++)
        printf("* "); //1 yıldız ve 1 bosluk
}
```

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

```
typedef struct
```

```
{
    char sirinlsim[100];
    double boy;
} smurf;
int main()
{
    smurf x;
    x = (smurf)
    {
        .boy=0.000052, .sirinlsim = "Mutlu"
    };
    printf("%.2lf\n", x.boy);
    printf("%s\n", x.sirinlsim);
    return 0;
}
```

0	.	0	0																
M	u	t	l	u															

////////////////////////////////////

AŞAĞIDAKİ ÇIKTI SOLDAKİ SORUYA AİTTİR.

Boyut : 2

```
* *
*
* *
```

Boyut : 4

```
* * * *
* * *
* *
* *
* * *
* * * *
```

Boyut : _

Kullanıcıdan bir pozitif tam sayı alan ve bu genişlikte, örnek çıktıdaki gibi, bir şekli ekrana yazdıran ve sayı alıp şekil yazdırmayı sürekli tekrar eden bir kod. Kod çalıştırıldıktan sonra kullanıcı sırasıyla "2" ve "4" girişleri yapmış ve her girişinden sonra enter'a basmıştır. Her enter'a bastığında ilgili boyutta(genişlik değerinde) bir şekil ekrana yazdırılmıştır. En sonunda kod sonlanmamış ve kullanıcıdan yeni değer istemiştir.

ÖĞRENCİ İSMİ:

SÜRENİZ 10 DAKİKADIR.

DERS NO(3 basamaklı):

BİL141 KISA SINAV VI ŞUBE 2 GRUP E SORULARI

- ASİSTANLARIN SINAV ESNASINDAKİ TEK GÖREVİ SINIFI GÖZETLEMektir. SORULARDAN PUAN ALABİLMEİNİZ İÇİN TAM OLARAK CEVAP VERMEİNİZ GEREKMEKTEDİR. KİSMİ PUANLAMA YAPILMAMAKTADIR. Örneğin, cevap 14.3 ise 14.30 yazan öğrenci o sorudan puan alamaz! / ve \ farkı, ' ve " farkı, $\leq$ yerine $\leq$ gibi detaylar da dâhil olmak üzere CEVABINIZIN TAM OLARAK DOĞRULUĞUNA emin olunuz. Ancak .(nokta) yerine ,(virgül) koymak ya da C yerine c yazmak gibi durumlar, soruda özellikle ölçülenler arasında değilse , puan kaybına yol açmaz.
- EKRAN ÇIKTISI SORULARINDA KODLARIN ÇIKTISINI HER HÜCRE BİR KARAKTERİ TEMSİL EDECEK ŞEKİLDE KUTULARA YAZINIZ. Cevabınızın kutuya sığmadığı durumlar olursa, ilgili kısma ek hücreler çizerek cevabınızı belirtebilirsiniz. Hata içerdiğini düşündüğünüz sorularda (mantık hatası haricindekilerde) ne tür bir hata olduğunu aşağıdaki seçeneklerden seçerek cevap kutusunun içine (örn. "HATA1" şeklinde) yazınız VE nedenini cevap kutusunun altına ya da soruya yakın başka boş bir alana (örn. "Çünkü koddaki ... eksik." şeklinde) belirtiniz. HATA1:Kod hiç derlenemeyecektir. HATA2:Kod derlenecek ama çalışması esnasında hata verecektir. HATA3:Kod derlenecek ama çalışması esnasında sonsuz döngüye girecektir. Yukarıdaki ifadelere uymayan hatalarda (örneğin mantık hatası içerdiğini düşündüğünüz sorularda) hatasız sorularda olduğu gibi sadece ekran çıktısını yazınız, HATA# vs. yazmayınız.
- KODDA BOŞLUK SORULARINDA EKRAN ÇIKTISINI SAĞLAYACAK ŞEKİLDE BOŞLUĞU KULLANDIĞINIZ TÜM KARAKTERLERE ($\leq$ yerine $\leq$ gibi) DİKKAT EDEREK DOLDURUNUZ. Gereksiz komut/ifadeler eklemeyiniz; bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

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

```
typedef struct
```

```
{
```

```
    char sirinlsim[100];
```

```
    double boy;
```

```
} smurf;
```

```
int main()
```

```
{
```

```
    smurf x;
```

```
    x = (smurf)
```

```
{
```

```
    .boy=0.000079, .sirinlsim = "Sirine"
```

```
};
```

```
printf("%.2lf\n", x.boy);
```

```
printf("%s\n", x.sirinlsim);
```

```
return 0;
```

```
}
```

```
// kod.c dosyasi
```

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

```
int f(int a);
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    printf("K");
```

```
    printf("%d", f(argc));
```

```
    return 0;
```

```
}
```

```
int f(int a)
```

```
{
```

```
    if(a==1) return 5;
```

```
    else
```

```
        return f(a-2)*2;
```

```
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
```

```
kod.exe 141 dersi ve ogrettikleri
```

```
K 20
```

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

KODDA BOŞLUK DOLDURMA SORUSU

```
#include <stdio.h>
void triangle(int, int);
void ara (int);
void spaceStar(int, int);
int main()
{
    int boyut;
    while(1)
    {
        printf("Boyut:");
        scanf("%d", &boyut);
        triangle(boyut,0);
        printf("\n\n");
    }
    return 0;
}
void triangle(int n, int p)
{
    if(n==1)
    {
        spaceStar(n,p);
        return;
    }
    int i;
    spaceStar(n,p);
    printf("\n");
    triangle( n-1 , p+1 );
    printf("\n");
}
```

```
    ara(p);
    for(i=1; i<=n; i++)
        printf("* "); //1 yildiz ve 1 bosluk
}
void ara (int x)
{
    int i;
    for(i=1; i<=x; i++)
        printf(" "); //2 bosluk
}
void spaceStar(int n, int bosluk)
{
    int i;
    ara(bosluk);
    for(i=1; i<=n; i++)
        printf("* "); //1 yildiz ve 1 bosluk
}
```

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

```
int main(int argc, char* argv[])
```

```
{
    printf("%s", (argc%3==2) ? argv[2]:argv[1]);

    // %sden sonra \n yok

    printf("%d",argc);

    return 0;
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
kod.exe gunesli hava
```

g	u	n	e	s	l	i	3		

(Kullanıcı komut ekranına, verilen iki komutu girmiş ve her komuttan sonra enter'a basmıştır.)

////////////////////////////////////

AŞAĞIDAKİ ÇIKTI SOLDAKİ SORUYA AİTTİR.

Boyut:2

```
* *
*
* *
```

Boyut:4

```
* * * *
* * *
* *
*
* *
* * *
* * * *
```

Boyut: _

Kullanıcıdan bir pozitif tam sayı alan ve bu genişlikte, örnek çıktıdaki gibi, bir şekli ekrana yazdıran ve sayı alıp şekil yazdırmayı sürekli tekrar eden bir kod. Kod çalıştırıldıktan sonra kullanıcı sırasıyla "2" ve "4" girişleri yapmış ve her girişinden sonra enter'a basmıştır. Her enter'a bastığında ilgili boyutta(genişlik değerinde) bir şekil ekrana yazdırılmıştır. En sonunda kod sonlanmamış ve kullanıcıdan yeni değer istemiştir.