

Adı Soyadı:

Numara:

Sınav süresi 80 dakikadır. Cep telefonlarınızı kapatınız. Sınav sorumlularının talimatlarına uyunuz. Sınav başlangıcından itibaren ilk 15 dakikada sınavı terk etmeyiniz. Soruların tamamı cevaplanacaktır.

1	2	3	4	Toplam

**SORULAR**

S.1. (20p) Aşağıda C benzeri bir kod verilmiştir. Bunara göre foo() metodunun değer, referans ve isim ile çağırma sonucunda ekran çıktılarını sebebiyle birlikte yazınız?

```
int i = 2;
void foo(int f, int g) {
    f = f + g;
    g = g + 4;
}
int main() {
    int a[] = {1, 1, 1, 1};
    foo(i, a[i-2]);
    printf("%d %d %d %d %d\n", i, a[0], a[1], a[2], a[3]);
}
```

S.2. (15p) a) Aşağıdaki context free gramer pascal benzeri bir dilin sentaksını göstermektedir. Terminal olmayan semboller büyük harf ve terminal semboller ise küçük harf ile gösterilmiştir. VAR değişken ismini ve CONST sabitli ifade etmektedir. Buna göre aşağıda verilen program parçasını üreten ayrıştırma ağacını çiziniz?

PROGRAM → procedure STMT-LIST

STMT-LIST → STMT STMT-LIST | STMT

STMT → do VAR = CONST to CONST begin STMT-LIST end | ASSN-STMT

procedure

do i = 1 to 100 begin

ASSN-STMT

ASSN-STMT

end

ASSN-STMT

(15p) b) 0 ve 1 üzerinde tanımlı bir Context free Gramer S, A ve B gibi üç terminal olmayan sembole sahiptir. Bu gramerin First ve Follow kümeleri aşağıda verilmiştir. Bunara göre bu gramerin kurallarını veriniz?

Sembol	First	Follow
S	{0,1}	{0,S}
A	{λ}	{0,1}
B	{λ}	{0,1}

(30 p) S. 3. Perl dilinde değişkenin önündeki "my" veya "local" keyword leri, o değişkenin statik(my) veya dinamik(local) scope özelliğinde olduğunu vurgulamaktadır. Bu bilgi doğrultusunda aşağıdaki Perl programının ekran çıktısını yandaki boşluğa yazınız

```
$x = 5;
$y = 5;
print $x,$y "\n";
&ana;
print $x,$y "\n";
sub ana {
    local $x = 10;
    my $y = 20;
    print $x,$y "\n";
    &alt;
    print $x,$y "\n";
}
sub alt {
    my $x;
    local $y;
    $x ++;
    $y ++;
}
```

5, 5  
11, 5  
10, 20  
5, 21  
5, 5

(20p) S.4. Aşağıdaki program parçalarının optimize edilmiş hallerini yandaki boşluklara yazınız.

```
for ( i=0; i<strlen(s); i++ ) {
    s[i] +=5;
}
```

```
for ( i=0; i<100; i++ ) {
    fonksiyon1(i);
}
```

```
int kup_alma(int a) {
    return a*a*a;
}
```

```
int global;
void f() {
    int i;
    i=1;
    global = 1;
    global = 2;
    return;
    global=3;
}
```

Başarılar.

Dr. Öğr. Üys. Erkan DUMAN, Doç. Dr. İlhan AYDIN

CEVAPLAR

1

2

3

4

5

6

7

8

9

10

11

12

13

# PROGRAMLAMA DİLLERİ ÇIKMIŞ DOKÜMAN

- 1) Identifier ( $\langle id \rangle$ ) kümesi =  $\{A, B, C\}$  olan sadece toplama ile çarpma işlemlerini içeren atama kurallarından oluşan bir dilin gramerini yazınız.  
 $A = A + B * C$  atama işlemini yazılan gramer kuralından türetin.  
(NOT: Çarpma işlemi öncelikli)

Cözüm

$$A = A + B * C$$

Gramer

$$\langle assign \rangle \rightarrow \langle id \rangle = \langle expr \rangle$$

$$\langle id \rangle \rightarrow A | B | C$$

$$\langle expr \rangle \rightarrow \langle id \rangle * \langle expr \rangle | \langle id \rangle + \langle expr \rangle | \langle expr \rangle | \langle id \rangle$$

Solo Dayalı Türetme

$$\langle assign \rangle \rightarrow \langle id \rangle = \langle expr \rangle$$

$$\rightarrow A = \langle expr \rangle$$

$$\rightarrow A = \langle id \rangle + \langle expr \rangle$$

$$\rightarrow A = A + \langle expr \rangle$$

$$\rightarrow A = A + \langle id \rangle * \langle expr \rangle$$

$$\rightarrow A = A + B * \langle expr \rangle$$

$$\rightarrow A = A + B * \langle id \rangle$$

$$\rightarrow A = A + B * C //$$

- 2) Darboğaz nedir? Açıklayınız.

Cevap:

Bir bilgisayarın belleği ve işlemcisi arasındaki bağlantı hızı o bilgisayarın hızını belirler. Program komutları çok hızlı, bağlantı hızından çok daha hızlı yürütülebilir; bağlantı hızı bu yüzden darboğaz sebep olur.

- 3) Break ve continue komutları arasında ne fark vardır?

Cevap:

\* Break komutu ile switch sonuna gidilir, oradan default çıkışa erişilir direk döngü bitirilir.

- Ama continue döngüyü bitirmez, kontrol kısmına gönderir.

- 4) Eğer 5 bitlik mimaride işaretli sayılar ikiye tümler yöntemi ile temsil ediliyorsa -15 ile +16 sayılarının toplama işleminin nasıl gerçekleştirildiğini gösteriniz.

Cözüm:

$$\begin{array}{r} 15 \rightarrow 01111 \\ 16 \rightarrow 10000 \\ + \\ -15 \rightarrow 10001 \end{array}$$

$$\begin{array}{r} 16 \rightarrow 10000 \\ -15 \rightarrow 10001 \\ + \\ \hline 10001 \\ 1 + 00001 \end{array}$$

- 5) Kendisine parametre olarak gelen 2 tane de $\pi$ skenin i $\pi$ eriklerini yer de $\pi$ istiren bir C veya C++ alt programı yazınız. Alt program colismasını bitirip, ça $\pi$ ıran programa d $\pi$ nd $\pi$ ld $\pi$ ğ $\pi$ nde yapılan de $\pi$ i $\pi$ iklik gecerli olmalıdır. Hem alt programı hem de nasıl ça $\pi$ ırılacağını kodlayınız.  
(NOT: C veya C++ default parametre ge $\pi$ cirme y $\pi$ ntemi de $\pi$ erile ça $\pi$ ırılmadır).

Cevap:

```
① void swap2 (int *a, int *b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}  
  
swap2 (&c, &d) ;
```

```
② [ ana program () {  
    int a = 3;  
    int b = 4;  
    swap (a, b);  
  
    swap (int c, int d) {  
        int temp = c;  
        c = d;  
        d = temp;  
    }  
}
```

} Referans ile  
ça $\pi$ ırma y $\pi$ ntemi

6)  $k = (j+13)/27$

loop :

if  $k > 10$  then goto out

$k := k + 1$

$i := 3 * k - 1$

goto loop

out : ...

Cevap:

for ( $k = (j+13)/27$  ;  $k \leq 10$  ;  $k = k + 1$ ) {

$i = 3 * k - 1$

}

Program par $\pi$ ca $\pi$ ıpına kar $\pi$ ıtlık  
gelen d $\pi$ ng $\pi$  y $\pi$ apısını C/C++  
veya Java kodunda yazınız.

for ( $k = (j+13)/27$  ;  $k > 10$  ;  $k = k + 1$ ) {  
 $i = 3 * k - 1$ ;  
}

- 7) C dilinde aşağıdaki fun isimli bir fonksiyon ve bu fonksiyonun kullandığı main programı tanımlanmıştır. a ve b sıktındaki öncelik durumlarında sum1 ve sum2 değişkenlerinin değerlerini hesapla.

Cevap

```
int fun(int *k){
    *k += 4;
    return 3*(*k) - 1;
}
```

a) Soldan sağa doğru

$$\text{sum1} = (10/2) + 4 = 46$$

$$\text{sum2} = 48$$

```
void main(){
```

```
int i = 10, j = 10, sum1, sum2;
```

b) Fonksiyon referansı öncelikli

```
sum1 = (i/2) + fun(i);
```

$$\text{sum1} = 48$$

```
sum2 = fun(&j) + (j/2);
```

$$\text{sum2} = 48$$

```
}
```

- 8) Aşağıdaki pascal program paracacığına karşılık gelen kodu if yapısı kullanarak C, C++ veya Java dillerinden birisinde yazınız.

```
case index - 1 0;
```

```
2, 4 : even := even + 1;
```

```
1, 3 : odd := odd + 1;
```

```
0, zero := zero + 1;
```

```
else error := true
```

```
end
```

Cevap

```
input = index - 1;
```

```
if (input == 2 || input == 4)
```

```
    even += 1;
```

```
else if (input == 1 || input == 3)
```

```
    odd += 1;
```

```
else if (input == 0)
```

```
    zero += 1;
```

```
else
```

```
    error = true;
```

- 9) C dilinde aşağıdaki gibi bir program paracacığı yazılmıştır. Kod çalıştıktan sonra j değişkeninin olacağı değeri hesaplayınız.

```
j = -3;
```

```
for (i = 0; i < 3; i++) {
```

```
    switch (j + 2) {
```

```
        case 3:
```

```
            case 2: j--; break;
```

```
            case 0: j += 2; break;
```

```
            default: j = 0;
```

```
        if (j > 0) break;
```

```
        j = 3 - i;
```

```
    }
```

Cevap

i	j
0	0
0	3
1	0
1	2
2	0
2	1

J = 1

10) İmperatif dillerin temel özellikleri nelerdir?

Cevap:

- Değişkenler
- Atama ifadeleri
- İterasyon

11) Bir programlama dilini değerlendirirken hangi kriterlerden bahsetmek gerekir? C ve Java dillerini bu kriterlere göre kıyaslayınız.

Cevap:

- \* Okunabilirlik (Readability): Programın okunabilme ve anlaşılabilme kolaylığı.
- \* Yazılabilirlik (Writability): Bir dilin program yazmada kullanılabilirliği.
- \* Güvenilirlik (Reliability): Sorunla uğraşma, sorunları çözme.
- \* Maliyet (Cost): En son toplam maliyet.
- \* Ortogonalite: Elimizdeki yapılarla kombinasyonlarla birleştirip yeni yapılar oluşturulabilir.

	Java	C
Okunabilirlik	+	+
Yazılabilirlik	+	+
Güvenilirlik	+	-
Maliyet	+	-
Ortogonalite	+	+

12) Derleyici ile yorumlayıcı arasındaki farklar?

Cevap:

Yorumlayıcı

- Tüm programı derym derym denetler.
- Bir döngü içindeki tüm deyimler her defasında çevrilir.

Derleyici

- Tüm programı bir kerede denetler.
- Sözdizimsel hataları bulur.
- Hata yok ise, programı nesne koda çevirir.
- Nesne kod daha sonra çalışabilir, koda çevrilir.

13) Belirleyici ne demektir? Açıklayın.

Cevap:

- Derleyici (compiler) anında dizinin başını ve sonunu hafızada belirlemesi için tutan yapıdır.

14) Terimlerin açıklmalarını yapınız.

3

Scope  $\Rightarrow$  (Komutların içindeki alandır)

- Bir değişkenin kapsamı değişkenin yürütülebilir olduğu komutların içindeki alandır.

Lifetime  $\Rightarrow$  (Koparıma Süresi)

- Bir değişkenin belirli bir bellek hücresiyle bağlanması ile bu bağın koparılması arasında geçen süredir.

Type Checking  $\Rightarrow$  (Tip hatalarının testi)

- Programın çalışması veya derlenmesi süresince verilen programın tip hatalarının test edilmesi işlemidir.

Binding  $\Rightarrow$  (Eleman arası ilişki (bağlama))

- Bir özellik, bir program elemanı arasında ilişki kurulması (binding) bağlama denir. Bağlama zamanı statik ve dinamik olması ile çeşitlilik gösterir.

\*(Memory leak)

15) Bellek sızıntısı ve Sallanma Pointer kavramlarını bir örnek üzerinde açıkla.

Cevap:

Dangling Pointer (Sallanma Göstergesi)

- Bir gösterge değişkeninin gösterdiği adreste geçerli veri olmaması durumunda, göstergenin serbest bırakılmış bir dinamik yapıya değistene işaret etmesi ile oluşur, serbest bırakılmış bellek adresini gösteren değistene dangling pointer denir.

ör

```
int n = 11;
int *a, *r, *h;
a = &n; r = NULL;
h = (int*) malloc(sizeof(int));
*h = 33;
r = h;
free(h);
```

r = boştu

h = 33 iken 33  $\rightarrow$  r'ye geçti,  
bu durumda h boşta kaldı.

Bellek Sızıntısı (Memory Leak)

- Bir programdaki dinamik bellek kullanımı mantıksal hataya istinaden oluşan bir bellek erişim arızası. Belleğin kullanımı bitmesine rağmen, belleğin serbest bırakılmamasına sebebiyet verir. Sonuçta bellek yetersizliği sebebiyle program sonlanır.

```
int n = 11;
int *a, *r, *h;
a = &n; r = NULL;
h = (int*) malloc(sizeof(int));
*h = 33;
h = NULL;
```

r boşta idi, h boş oldu ama 33 değeri gitti. bellek sızıntısı.

Çözüm için yöntemler

- 1) Yazıt = Ek bellek hücresidir.
- 2) Kilit ve Anchor = Gösterici çiftinden oluşur.

16) Statik ve dinamik tip bağlamasını birer cümle ile açıklayıp dil örneği veriniz?

Cevap:

Statik Tip Bağlama: Tiplerin isimlerle derleme zamanında bağlandığı diller olarak nitelendirilir. Burada derleyici, tip hatalarını, program çalıştırılmadan önce yakalar.

Dinamik Tip Bağlama: Bir programlama dilinde bir değişkenin tipi çalışma zamanında, değişkenin bağlandığı değer ile belirlenir.

17) Fonksiyonel yan etti kavramını bir örnek üzerinde gösteriniz.

Cevap:

Bir fonksiyon iki yönlü bir parametreyi veya lokal olmayan bir değişkeni depistirdiğinde meydana gelir.

- Bir ifadede çağrılmış bir fonksiyon ifadenin başka bir operantını depistirdiğinde ortaya çıkar.

```
int fun (int *u) {
```

```
    *u = *u/2 ;  
    return *u ; }
```

```
o = 10 ;  
b = o + fun(o) ;  
// fun, parametresini  
depistiriyor
```

2 muhtemel çözüm

1) Dil tanımlaması yapılır. İki yönlü parametre ve global depistirme olmayarak.

2) Operantların işlem sırasını belirtmek için dil tanımlaması yapılır.

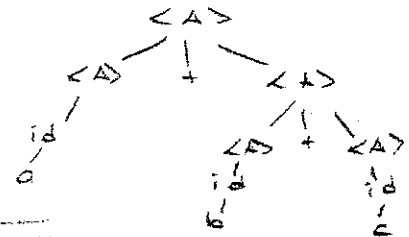
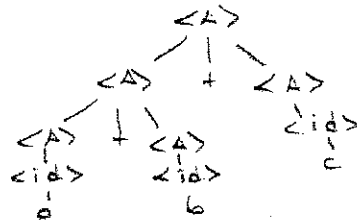
18) Aşağıdaki gramerin belirsiz olup olmadığını ispatla.

Cevap:

$\langle S \rangle \rightarrow \langle A \rangle$

$\langle A \rangle \rightarrow \langle A \rangle + \langle A \rangle \mid id$

$\langle id \rangle \rightarrow a \mid b \mid c$



İki farklı türetme ağacı oluşturulmuş ve belirlenmiştir.

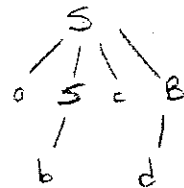
19) Aşağıdaki gramer ile abcd ve acccbcd ifadelerinin türetilip türetilmeyeceğini parse ağacı çizerek gösteriniz.

Cevap:

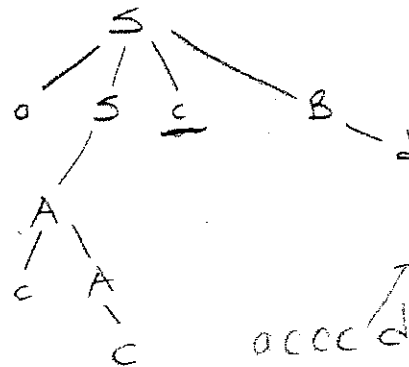
$S \rightarrow aScB \mid A \mid b$

$A \rightarrow cA \mid c$

$B \rightarrow d \mid A$



\* abcd oluşur.



\* acccbcd oluşmaz.

b için S'ın başına  
başlamaz.



20) program A() {

x, y, z : integer;

procedure B()

{

y : integer;

y = 0;

x = z + 1;

z = y + 2;

procedure C()

{

z : integer;

procedure D()

{

x : integer;

x = z + 1;

y = x + 1;

call B();

}

z = 5;

call D();

}

x = 10;

y = 11;

z = 12;

call C();

print x, y, z;

### STATİK KAPSAM

(x, y, z)	(y)	(z)	(x)
A	B	C	D
x = <del>10</del> 13	y = 0	z = 5	x = 6
y = <del>11</del> 7			
z = <del>12</del> 2			

x = 13      y = 7      z = 2

### DİNAMİK KAPSAM

(x, y, z)	(y)	(z)	(x)
A	B	C	D
x = 10	y = 0	z = <del>5</del> 2	x = 6 <del>14</del>
y = <del>11</del> 7			
z = 12			

x = 10      y = 7      z = 12

21)

function sub(int a, int b, int c)

begin

b := a + c;

a := c + 1;

print a, b, c;

end

function main

begin

int i := 3;

int j := 5;

int k := 8;

sub(i, j, j + k);

print i, j, k;

end

Call by value

14	16	13
3	5	8

Call by result

14	16	13
14	16	8

Call by reference

14	16	13
14	16	13

Call by name

14	16	24
14	16	8

a	b	c
1	1	1
1	1	1
2	5	13

\* k'yi tutan bir adres yok  
(değişken değeri)

22) a) 3 nonterminal, 4 terminal ve 5 kural sahip bir CFGn grameri açıklayarak yaz.

$A \rightarrow aB \mid cd$

$B \rightarrow a \mid d$

$C \rightarrow b$

$A, B, C \}$  Non-terminaller

$a, b, c, d \}$  terminaller

b) Lexical analiz ve Syntax analiz kavramlarını birer örnek üzerinden açıklayınız.

Syntax Analiz: Bir bilgisayar programının yapısını irdelleyen ya da belirli bir programlama dilinin sözdizimine uyup uymadığını sınavan analiz.

Lexical Analiz (metinsel): Bir derleyicinin ön ucunda yer alan metinsel çözümleyici, bir kaynak programı bir dizi token'a çevirir.

c) Gramerlerde belirsizlik ne demektir. Belirsiz olmayan gramer için örnek bir gramer yazın.

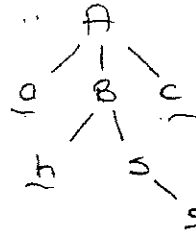
- Bizden istenilen sonucu belirli bir gramerle ifade iki şekilde veya daha fazla şekilde türetebiliyorsak, bu gramer belirsizdir.

Belirsiz olmayan gramer

$A \rightarrow aBc$

$B \rightarrow hS$

$S \rightarrow d$



ahcd'yi oluştur.

$\Leftarrow$

23) BNF kural kümemizi;

Kural 1  $\Rightarrow \langle ifade \rangle ::= \langle ifade \rangle \text{ or } \langle terim \rangle \mid \langle terim \rangle$

Kural 2  $\Rightarrow \langle terim \rangle ::= \langle terim \rangle \text{ and } \langle faktor \rangle \mid \langle faktor \rangle$

Kural 3  $\Rightarrow \langle faktor \rangle ::= \text{not } \langle faktor \rangle \mid (\langle ifade \rangle) \mid \text{true} \mid \text{false}$

Programlayıcı kod içerisine "not (true or false)" ifadesi yazmış ise bu ifade için parse tree veya kural kümesinden türetme yaparak düzenli bir ifade olup olmadığını analiz et.

çözüm

$= \langle faktor \rangle$

$= \text{not } \langle faktor \rangle$

$= \text{not } (\langle ifade \rangle)$

$= \text{not } (\langle ifade \rangle \text{ or } \langle terim \rangle)$

$= \text{not } (\langle terim \rangle \text{ or } \langle terim \rangle)$

$= \text{not } (\langle faktor \rangle \text{ or } \langle terim \rangle)$

$= \text{not } (\langle faktor \rangle \text{ or } \langle faktor \rangle)$

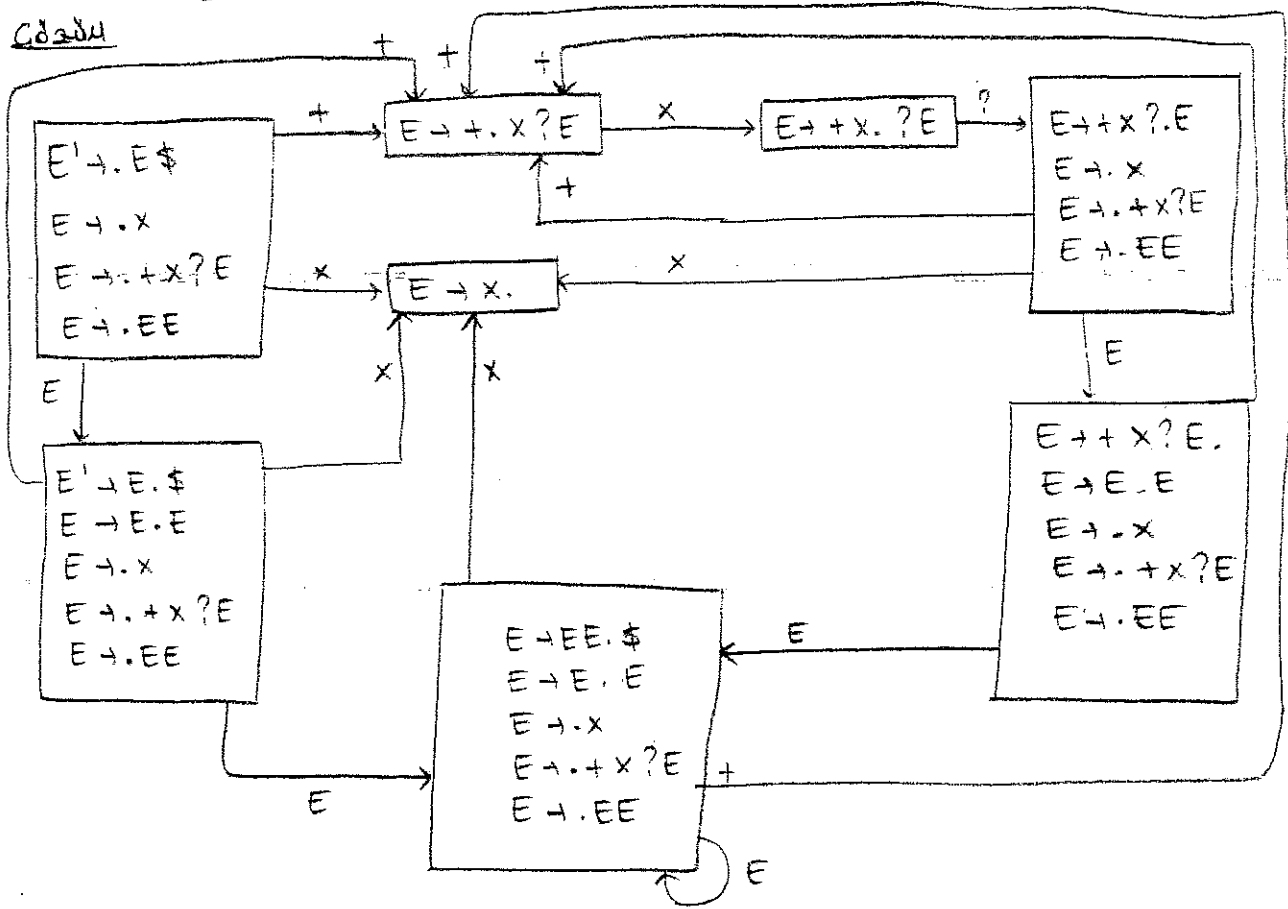
$= \text{not } (\text{true or } \langle faktor \rangle)$

$= \text{not } (\text{true or false}) //$

24) Aşağıdaki CFG için LR(0) parsing DFA diyagramını açıklayarak çiziniz.

$E \rightarrow x \mid +x?E \mid EE$

Çözüm



25) Aşağıdaki CFG için First ve Follow kümelerini açıklayarak elde ediniz.

Çözüm

$S \rightarrow ABD$

$A \rightarrow a \mid BSB$

$B \rightarrow b \mid D$

$D \rightarrow d \mid \epsilon$

	First	Follow
S	a, b, d, $\epsilon$	\$, b, d
A	a, b, d, $\epsilon$	b, d
B	b, d, $\epsilon$	d, \$, b, a
D	d, $\epsilon$	\$, b, d



## SINAV SORUSU

Aşağıdaki program parçalarının optimize edilmiş hollerini yandaki boşluklara yazınız.

```
for (i=0; i<strlen(s); i++){  
    s[i] += 5;  
}
```

strlen değişken gibi bir şey tanımladığı için bu kod optimize edilemez.

Baska cevap → 

```
int uzunluk = strlen(s);  
int i;  
for (i=0; i<uzunluk; i++){  
    s[i] += 5;  
}
```

```
for (i=0; i<100; i++){  
    fonksiyon1(i);  
}
```

```
int i;  
int a = 100;  
for (i=0; i<a; i++){  
    fonksiyon1(i);  
}
```

```
int kup_alma(int a){  
    return a*a*a;  
}
```

```
int kup_alma(int a){  
    x = a*a;  
    a = x*a;  
    return a;  
}
```

Baska cevap

→ 

```
int kup_alma(int a){  
    int kup = a*a*a;  
    return kup;  
}
```

```
int global;  
void f(){  
    int i; } Koku kısaltma  
    i = 1;
```

```
int global;  
void f(){  
    int i = 1;  
    global = 2;  
}
```

```
global = 1; i=1 tanımladığı için  
global = 2; global=1 hiçbir işlev  
arada yapılmaması gerekir
```

```
return;
```

```
global = 3;  
}
```

Kabul oluncaya kadar  
çünkü en son return  
değer

Baska cevap → 

```
int global;  
void f(){  
    global = 2;  
    return;  
}
```

Şimdi context free gramer pascal benzeri bir dilin sentaksini göstermektedir. Terminal olmayan semboller büyük harf ve terminal semboller ise küçük harfler ile gösterilmiştir. VAR Depisten ismini ve CONST sabiti ifade etmektedir. Buna göre aşağıda verilen program parçasını üreten ayrıştırma ağacını çiziniz?

PROGRAM  $\rightarrow$  procedure STMT-LIST

STMT-LIST  $\rightarrow$  STMT STMT-LIST | STMT

STMT  $\rightarrow$  do VAR = CONST to CONST begin STMT-LIST end | ASSN-STMT

procedure

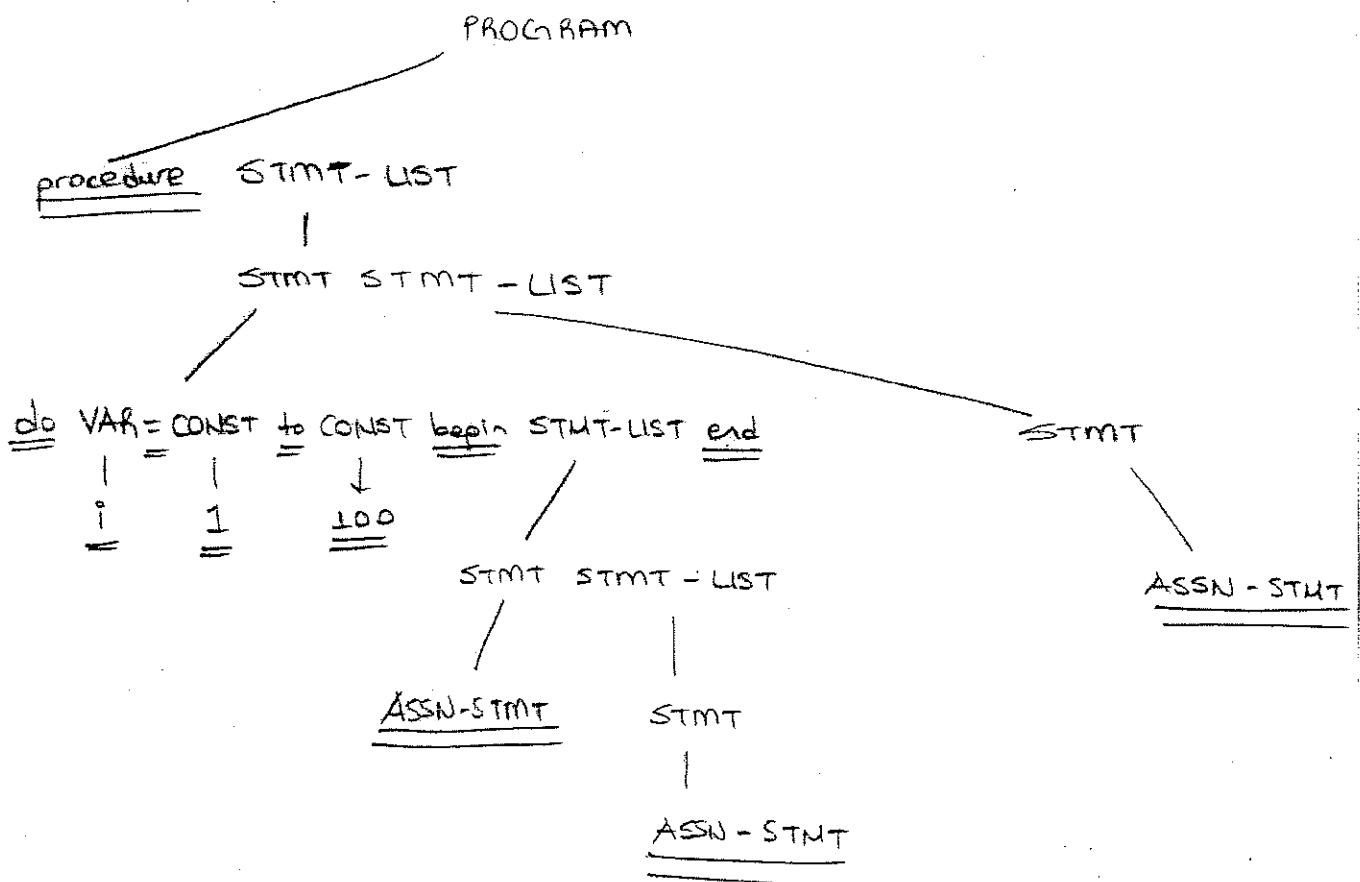
do i = 1 to 100 begin

ASSN-STMT

ASSN-STMT

end

ASSN-STMT



## FIRST - FOLLOW SINAV SORUSU

0 ve 1 üzerinde tanımlı bir Context free Gramer  
S/A ve B gibi  $\lambda$  terminal olmayan sembollere sahiptir.  
Bu gramerin First ve Follow kümeleri aşağıda verilmiştir.  
Şu soru göre bu gramerin kurallarını veriniz?

<u>Sembol</u>	<u>First</u>	<u>Follow</u>
S	{0,1}	{0,1,\$}
A	{ $\lambda$ }	{0,1}
B	{ $\lambda$ }	{0,1}

Cevap

~~$S \rightarrow 011 \mid AS \mid BS$~~   
 ~~$A \rightarrow \lambda$~~   
 ~~$B \rightarrow \lambda$~~

$S \rightarrow 011 \mid SO$

$A \rightarrow \lambda \mid AS$

$B \rightarrow \lambda \mid BS$

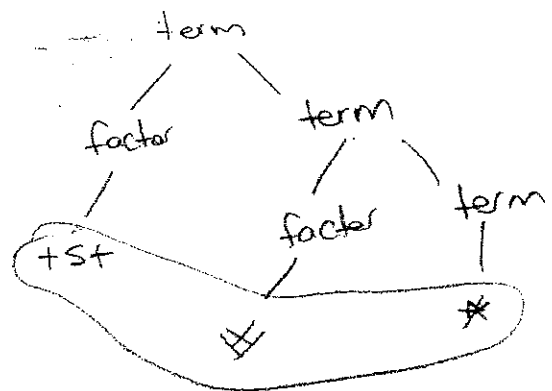
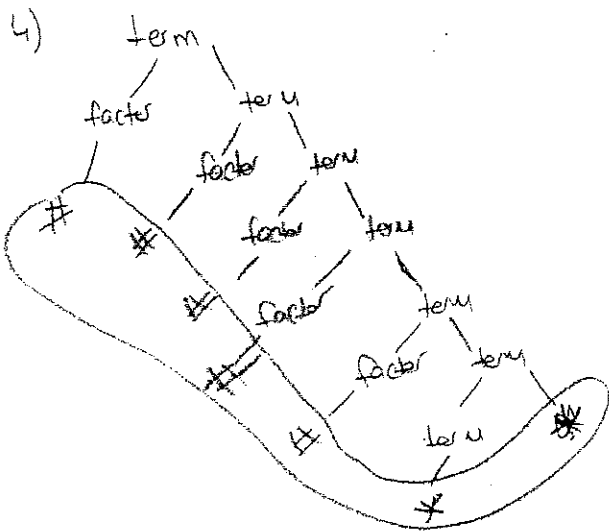
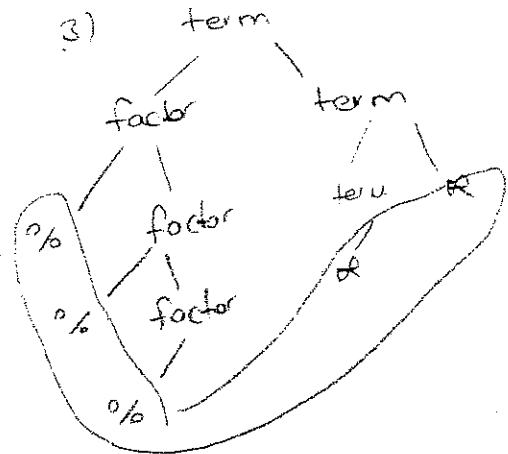
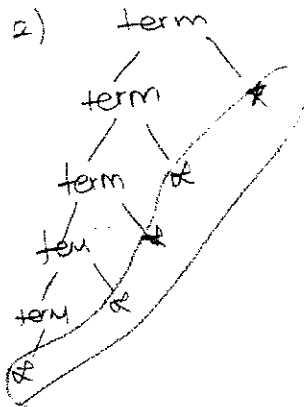
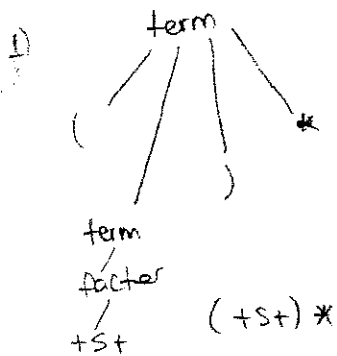
Soluyor.  $\rightarrow$

	<u>First</u>	<u>Follow</u>
S	0,1	0,1,\$
A	$\lambda$	0,1
B	$\lambda$	0,1

$\langle \text{term} \rangle \rightarrow \dots$   
 $\langle \text{factor} \rangle \rightarrow \{ '+s+' \mid '\#'\} \mid '\%' \langle \text{factor} \rangle$

input string	Legal or Not legal
*	Legal
1) (+\$+*)	Not legal
2) * ** *	Legal
3) % % % **	Legal
4) * * * * *	Legal
5) + \$+ * *	Legal

Cözüm





# Programlama Dilleri Çıkmış Sorular

11) 20 puan

13)

3

Terimlerin açıklamalarını yapınız?

① Scope → Komutların içindeki alandır.  
Bir değişkenin kapsamı, değişkenin yürütülebilir olduğu komutların içindeki alandır.

② Lifetime koparılma süresi

Bir değişkenin belirli bir bellek hücresiyle bağlanması ile bu bağın koparılması arasında geçen süredir.

③ Type Checking Tip hatalarının testi

Programın çalışması veya derlenmesi süresince verilen programın tip hatalarının test edilmesi işlemidir.

④ Binding → eleman arası ilişki (bağlama)  
Bir özellikte, bir program elemanı arasında ilişki kurulması binding (bağlama) denir.  
Bağlama zamanı statik ve dinamik olması ile çeşitlilik gösterir.

12) 25 puan

Bir programlama dilini değerlendirirken hangi kriterlerden bahsetmek gerekir?  
C ve Java dillerini bu kriterlere göre karşılaştırınız.

Eğer bir programlama dilinde dizilerin hafızaya yerleşimi satır öncelikli ise,

float matris [10] [5];

şeklinde deklar edilen diziye 2 satır 2 sütundaki elemanın hafızadaki adresi nasıl hesaplanır?

Sıfıncı, indisten başlandığı için

1. elemanı 2. satır ve 1. elemanı 2. sütundadır.

2. 1. - - -  
- 0 - - -  
- - - - -  
- - - - -  
- - - - -

14) 15 Puan

Bir programlama dilinde; birden fazla alt programda

int x;

şeklinde bir değişken tanımlanmış ise kod içerisinde x ile hangi değişkener erişim yapılacağına nasıl karar verilir?

Programlama dilinin hangi özelliği belirleyicidir? Alt programlar içerisinde alt program tanımlayabildiğini de göz önüne alınız.

	Java	C
Okunabilirlik	✓	✓
Yazılabilirlik	✓	✓
Güvenilirlik	✓	-
Maliyet	✓	-
Ortogonalite	✓	-

Okunabilirlik → prog- anlayabilme kolaylığı

Yazılabilirlik → Sayutlama, Anlamlılık

Güven → sızlara uygunluk → Type checking

Maliyet → En son toplam maliyet

Ortogonalite → Eldeki yapılarla kombinasyonları

# 10.03.2021 11.11.16

15 20 puan

16 10 puan

Dangling Problem nedir? Dangling probleminin oluşturmağı bir durum örneğı veriniz. Gözüm için kullanılabilecek bir yöntemi izah ediniz.

Betimsleyici ne demektir Açıklayınız  
compiler anında dizinin başını ve sonunu hafızada betiklemesi için tutan yapısı,

Dangling Pointer (Sallanan Gösterge)

Bir gösterge değişkeninin gösterdiği adreste geçerli veri olmaması durumunda, göstergenin serbest bırakılmış bir dinamik yığın değişkene işaret etmesi ile oluşur. Serbest bırakılmış bellek adresini gösteren değişkene dangling pointer denir.

Dangling Pointer örneğı

```
int n=43;  
int *a, *r, *h;  
a=&n; r=NULL;  
h=(int*) malloc(sizeof(int));  
*h=33;  
r=h  
free(h);
```

r = boştu.  
h=33 iken 33'e gitti,  
bu durumda h boşta kaldı.

2) Bellek Sızıntısı örneğı

```
int n=11;  
int *a, *r, *h;  
a=&n; r=NULL;  
h=(int*) malloc(sizeof(int));  
*h=33;  
h=NULL;
```

r boşta idi, h boş oldu ama 33 çöpe gitti; bellek sızıntısı.

Gözüm için yöntemler

1) Varlık = Ek bellek hücresidir;

2) Kilit ve Anahtar = Gösterici çiftinden oluşur;

(4)

## Çıkış Soruları

① (25 Puan)

```
j = 3
for (i = 0; i < 3; i++) {
    switch (j + 2) {
        case 3:
        case 2: j--; break;
        case 0: j = 2; break;
        default: j = 0;
    }
}
```

if (j > 0) break;

j = 3 - i;

}

Kod çalıştıktan sonra j değişkeninin alacağı değerleri hesaplayınız.

cevap

j = 1

1	3
0	3
1	0
1	2
2	0
2	1

③ 25 Puan

①

```
int fun(int *k) {
    *k = 4;
    return 3 * (*k) - 1;
}
```

```
void main() {
    int i = 10, j = 10, sum1, sum2;
    sum1 = (i/2) + fun(i);
    sum2 = fun(&j) + (j/2);
}
```

C dilindeki fun, fonksiyonunda sum1 ve sum2 değerlerini hesaplayınız.

② soldan\_sagaya doğru

sum1 = 46  
sum2 = 48

③ Fonksiyon referansı öncelikli

sum1 = 48

sum2 = 48

④ 25 Puan

k = (j + 13) / 23

loop:

if k > 10 then goto out

k = k + 1

i = 3 \* k - 1

goto loop

out: --

Pascal program parçasına karşılık gelen kodu if yapısı kullanarak C, C++, Java dillerinden birinde yazın

cevap

input = index - 1;  
if (input == 2 || input == 4)

even += 1;

else {

if (input == 1 || input == 3)

odd += 1;

else {

if (input == 0)

zero += 1;

else error = 1;

Program parçasına karşılık gelen C, C++ veya Java kodunu yazın

for (k = (j + 13) / 23; k < 10; k++) {

i = 3 \* k - 1;

}

□

5) 20 puan

Kendisine parametre olarak gelen 2 tane değişkenin içeriklerini yer değiştiren bir C veya C++ alt-programı yazınız. Alt program çalışmasını bitirip, çağıran programa döndüğünde yapılan değişiklik geçerli olmalıdır. Hem alt-programı hem de nasıl çağırılacağını kodlayınız.

(NOT: C veya C++ default parametre geçirme yöntemi değer ile çağırır).

void swap2(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

} swap2(8, 8d) }

Referans ile

2)

anaprogam() {

int a=3;

int b=4;

swap(a, b); }

swap(int c, int d) {

int temp = c;

c = d;

d = temp; }

6) 15 puan

Eğer 5 bitlik mimaride işaretli sayılar ikiye tamlar yöntemi ile temsil ediliyorsa -15 ile +16 sayılarının toplama işleminin nasıl gerçekleştirildiğini gösteriniz.

$$\begin{array}{r} 15 \rightarrow 01111 \\ 10000 \\ + 1 \\ \hline -15 \rightarrow 10001 \end{array} \quad \begin{array}{r} 16 \rightarrow 10000 \\ + -15 \rightarrow 10001 \\ \hline 1 \rightarrow 10001 \end{array}$$

7) 15 puan

break ve continue komutları arasında ne fark vardır?

\*Break komutu ile switch sonuna gidilir, oradan default çıkışa erişilir direk. Döngüyü bitirir.

Fakat continue döngüyü bitirmez kontrol kısmına gönderir.

8) 10 puan

Darboğaz Bir bilgisayarın belleği ve işlemcisi arasındaki bağlantı hızı o bilgisayarın hızını belirler. Program komutları çoğu kez, bağlantı hızından çok daha hızlı yürütülebilir; bağlantı hızı bu yüzden darboğaza sebep olur.

9) 10 puan

İmperatif dillerin temel özellikleri

3 tane dir,

- Değişkenler
- Atama İfadeleri
- İterasyon

10) 30 puan

İdentifier (<id>) kümesi = {A, B, C} olan sadece toplama ile çarpma işlemlerini içeren atama kurallarından oluşan bir dilin gramerini yazınız. A = A + B \* C atama işlemini yazılan gramer kurallarından türetin. (NOT: Çarpma işlemi öncelikli)

Gramer

A = A + B \* C

<assign> → <id> = <expr>

<id> → A | B | C

<expr> → <id> \* <expr>

| <id> + <expr>

| <expr>

| <id>

Sola Dayalı Türetme

<assign> ⇒ <id> = <expr>

⇒ A = <expr>

⇒ A = <id> + <expr>

⇒ A = A + <expr>

⇒ A = A + <id> \* <expr>

⇒ A = A + B \* <expr>

⇒ A = A + B \* <id>

⇒ A = A + B \* C

2

örn! (Sınav Sorusu)

Soldan-Sağa doğru

```
int fun (int *k) {
```

sum1 = 46

```
    *k += 4;    *k = *k + 4;     $\frac{84}{2 \times 2}$ 
```

sum2 =

```
    return 3*(*k) - 1;
```

```
}
```

```
void main() {
```

$\frac{14}{1 \times 3}$   $\frac{41}{4 \times 2 - 1 = 41}$   $\frac{41}{1 \times 2}$

```
    int i = 10, j = 10, sum1, sum2;
```

```
    sum1 = (i/2) + fun(i);     $\rightarrow 5 + \text{fun}(10)$   
                                 $= 41 + 5 = 46$ 
```

```
    sum2 = fun(8j) + (j/2);     $\rightarrow \text{fun}(80) + 5$ 
```

```
}
```

örn! Break ve Continue komutları arasındaki fark!

Break komutu ile switch sonuna gidilir, oradan default çıkışına erizilir. Döngüyü bitirir.

Fakat continue döngüyü bitirmeye kontrol kısmına gönderir.

örn! Döngü nedir?

Bir bilgisayarın belleği ve işlemcisi arasındaki bağlantı hızı o bilgisayarın hızını belirler. Program komutları doğru kes, bağlantı hızından dolayı daha hızlı yürütülebilir. Bağlantı hızı bu yüzden döngüdür, sebep olur.

Örn imperative dillerin temel özellikleri?

3 tane dir

- Değişkenler

- Atomik ifadeler

- İterasyon

Örn identifier (<id>) kumesi {A,B,C} olan sadece  
top ve carpma is. gerceklestiren atomik kurallarda  
Olusan bir dilin gramerini yaziniz  $A = A + B * C$  atomik  
islemine yazilan gramer kurallarından türettin. (NOT: Carpma  
islemi öncelikli)

$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$\langle \text{id} \rangle \rightarrow A | B | C$

$\langle \text{expr} \rangle \rightarrow \langle \text{id} \rangle * \langle \text{expr} \rangle |$

$\langle \text{id} \rangle + \langle \text{expr} \rangle |$

$\langle \text{expr} \rangle |$

$\langle \text{id} \rangle$

GRAMER

→ SOLA DAYALI TURETME ←

$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$\Rightarrow A = \langle \text{expr} \rangle$

$\Rightarrow A = \langle \text{id} \rangle + \langle \text{expr} \rangle$

$\Rightarrow A = A + \langle \text{id} \rangle * \langle \text{expr} \rangle$

$\Rightarrow A = A + B * \langle \text{expr} \rangle$

$A = A + B * \langle \text{id} \rangle$

$A = A + B * C$

Örn: Aktivasyon kaydı Alt prog. değişkenlerin bulunduğu bölüm. Prog. birimini yürütmek için butun datayı içerir.

\* Dinamik Bağlımlı diller → Çevirmeye (interpretation) yöneliktir.

\* Statik " " → Dönüştürmeye (translation) yöneliktir.

Örn: Belirleyici ne demektir?  
Derleme anında dizinin başını ve sonunu belirleyerek belirlenmesi için tutan yapıdır.

Örn: Bellek Sızıntı Örneği:

int n = 11; → r boşta idi, h boş oldu

int \*a, \*r, \*h; ama 33 a'ye gitti;

a = 8n; r = NULL; bellek sızıntısı için yöntemleri

h = (int \*) malloc(sizeof(int)); 1) Yozıt = Ek bellek hücreleridir.

\*h = 33; h = NULL; 2) Kilit ve Anahtar = Gösterici c.p'ten alırsın

Örn: Bir prog. dilini değerlendirirken

hangi kriterlerden bahsetmek gerekir?

C ve Java dillerini bu kriterlere göre kıyaslayın!

	Java	C	
Okunabilirlik	✓	✓	→ Anlaşılabilirlik kolaylığı
Yazılabilirlik	✓	✓	→ Soyutlama, Anlatılabilirlik
Çevrilebilirlik	✓	—	→ Doğru uyumluluk Type checking İstatik işleme Farklı edildikçe
Maliyet	✓	—	→ En son toplam maliyet
Ortogonalite	—	—	→ Eski yapılarla kombinasyonlar

Örn=

Scope = Komutların içindeki alanlardır. Bir değişkenin yürütülebilir old. komutların içindeki alanıdır.

Lifetime = (Kapanılma Süresi) Bir değişkenin belirli bir bellek hücresiyle bağlanması ile bu bağın kapanması arasında geçen süredir.

Type Checking = (Tip hatalarının testi) prog çalışması veya derlenmesi sırasında verilen prog. tip hatalarının test edilmesi işlemidir.

Binding = Bir prog elemanı arasında ilişki kurulması binding (bağlama) denir. Bağlama zamanı statik ve dinamik olması ile esitlilik gös.

Built in type = Prog dilinin kendisinden var olan tiplerdir.

User defined type = Kullanıcı tarafından tanımlanan değişkenlerdir.

Link = (Information Binding) = Tipin özelliklerini tanımlıyarak kullanıcı için

BNF = Syntax tanımlamaya yarayan dildir. (metadildir.)

Abstraction = Soyutlama. Kullanıcı için o an çok gerekli olmayan bilgilerin soyutlanması

Static = Hafızadaki derleme zamanı sırasında yerli belirlenir ve sabitlenen değişkenler (int a=5)

Semi-Static = Fiziksel depozaya bağlanma, aktivasyon kaydının adresini tutar ve run-time anında tamamen boyutu onboson değiştirilir.

Semi-Dynamic = Boyutu tamamen runtime sırasında belli olan değişkenlerdir.



$$\begin{matrix} \text{referans} \\ \left\{ \begin{array}{l} \text{48} = \text{sum1} = \text{46} \\ \text{48} = \text{sum2} = \text{48} \end{array} \right\} \end{matrix} \quad \left. \begin{array}{l} \text{3} \\ \text{soldan - sağı} \end{array} \right\}$$

Descriptor (Tanımlayıcı) = Derleme zamanı içinde, dizinin başını ve sonunun hafızada belirlenmesi için tutan yapıdır.

Atomik = Hem boyutu, hem de kayıt tipi run-time'da bile belli olmayan dizilerdir.

Top = Down = @nce main'in yazılıp, sonra alt prog. yazılmasıdır.

Heap = Yığın tersine dolışın hafıza yapısıdır. Tamamen dinamik değişkenler buraya kayılır.

Garbage collector C#, Java

Kullanılmayan referansların, olmayan nesnelerin yok edilmesi

Prota =  
Avantaj = Hız  
Dezavantaj = Karmaşıklık

Ön:

Derleyici ve Yorumlayıcı arasında ne fark var?

Birbirlerine göre avantaj ve dezavantajlarını iletin edin!

Derleyici prog. makine diline çevirir, Yorumlayıcı ise yüksek sev. herhangi bir dili başka bir dile çevirir.

Yorumlayıcıda çevirici - yada prog. imlementasyonu daha kolay, daha yavaş yürütme daha fazla boş alana ihtiyaç, yüksek sev. dillerde daha nadir kullanılır, web dillerinde önem kazanır.





Adı Soyadı:

Numara:

Sınav süresi 80 dakikadır. Cep telefonlarınızı kapatınız. Sınav sorumlularının talimatlarına uyunuz. Sınav başlangıcından itibaren ilk 15 dakikada sınavı terk etmeyiniz. Soruların tamamı cevaplanacaktır.

1	2	3	4	Toplam

SORULAR

(15p) S.1.a) Sallanan işaretçi kavramını bir örnek ile açıklayınız, çözüm yöntemlerini veriniz?

b) Statik ve dinamik tip bağlamasını birer cümle açıklayıp, dil örneği veriniz?

c) Fonksiyonel yan etki kavramını bir örnek üzerinde gösteriniz?

(15p) S.2. Aşağıda verilen program için statik ve dinamik kapsama göre çıktıyı bulunuz?

Program A()

```
{
  x, y, z: integer;
  procedure B()
  {
    y: integer;
    y=0;
    x=z+1;
    z=y+2;
  }
  procedure C()
  {
    z: integer;
    procedure D()
    {
      x: integer;
      x = z + 1;
      y = x + 1;
      call B();
    }
    z = 5;
    call D();
  }
  x = 10;
  y = 11;
  z = 12;
  call C();
  print x, y, z;
}
```

b) Aşağıdaki gramerin belirsiz olup olmadığını ispatlayınız?

$\langle S \rangle \rightarrow \langle A \rangle$

$\langle A \rangle \rightarrow \langle A \rangle + \langle A \rangle \mid \langle id \rangle$

$\langle id \rangle \rightarrow a \mid b \mid c$

c) Aşağıdaki gramer ile abcd ve acccbd ifadelerinin türetilip türetilenemeyişini parse ağacı çizerek gösteriniz?

$S \rightarrow aScB \mid A \mid b$

$A \rightarrow cA \mid c$

$B \rightarrow d \mid A$

(50p) S.4. Aşağıdaki kod parçasında yapılan alt program çağırısı her şık için farklı bir teknik ile gerçekleştirilirse ilgili program çıktısı ne olur? (Ekran yazılan satırlardaki değerleri aşağıdaki tabloya yazınız.)

function sub(int a, int b, int c)

begin

b := a + c;

a := c + 1;

print a, b, c;

end

function main

begin

int i := 3;

int j := 5;

int k := 8;

sub(i, j, j + k);

print i, j, k;

end

call by value	call by result	call by reference	by	call by name

(20p) S.3. a) Aşağıdaki gramer kuralları için pairwise disjoint set tesini uygulayınız?

$S \rightarrow aSb \mid bAA$

$A \rightarrow b(aB) \mid a$

$B \rightarrow aB \mid a$

Başarılar.

Yrd. Doç.Dr. Erkan DUMAN

Doç. Dr. İlhan AYDIN

CEVAPLAR





Adı Soyadı:

No:

PROGRAMLAMA DİLLERİ  
BİLGİSAYAR MÜHENDİSLİĞİ

Sınav süresi 90 dakikadır. Cep telefonlarınızı kapatınız. Sınav sorumlularının talimatlarına uyunuz. Sınav başlangıcından itibaren ilk 15 dakikada sınavı terk etmeyiniz. Soruların tamamı cevaplanacaktır.

1	2	3	4	5	Toplam

## SORULAR

1. (15p) Aşağıdaki CFG için First ve Follow kümelerini açıklayarak elde ediniz.

$$S \rightarrow TS \mid [S]S \mid )S \mid \epsilon$$

$$T \rightarrow (X)$$

$$X \rightarrow TX \mid [X]X \mid \epsilon$$

2. (20p) Aşağıdaki CFG için LR(0) parsing DFA diyagramını açıklayarak çiziniz.

$$E \rightarrow E+T \mid T$$

$$T \rightarrow (E) \mid id$$

3. (15p) a) 3 nonterminal, 4 terminal ve 5 kurala sahip bir CFG grameri açıklayarak yazınız.  
b) Lexical analiz ve Syntax analiz kavramlarını birer örnek üzerinden açıklayınız.  
c) Gramerlerde belirsizlik ne demektir. Belirsiz olmayan gramer için örnek bir gramer yazınız.
4. (20p) Bir programlama dilinde aşağıdaki kurallardan oluşan bir BNF kural kümemiz olduğunu kabul edelim:

Kural 1  $\Rightarrow \langle ifade \rangle ::= \langle ifade \rangle or \langle terim \rangle \mid \langle terim \rangle$

Kural 2  $\Rightarrow \langle terim \rangle ::= \langle terim \rangle and \langle faktor \rangle \mid \langle faktor \rangle$

Kural 3  $\Rightarrow \langle faktor \rangle ::= not \langle faktor \rangle \mid (\langle ifade \rangle) \mid true \mid false$

Eğer programlayıcı kod içerisine "not (true or false)" ifadesi yazmış ise bu ifade için parse tree veya kural kümesinden türetme yaparak düzenli bir ifade olup olmadığını analiz ediniz.

5. (30p) a) Yorumlayıcı ile derleyici arasında farkı C ve JAVA dillerini kıyaslayarak anlatınız?

b) Programlama dillerinde soyutlama kavramını izah ediniz? İlk defa hem veri hem de işlev soyutlayan dil hangisidir?

c) Von-Neuman Mimaride döngü işlemi veya başka bir deyişle tekrarlama işlemi yapmak için Recursive kodlama neden iterasyon kurmaya göre dezavantajlıdır?

Başarılar Dileriz.

Doç.Dr. Mehmet KARAKÖSE

Yrd.Doç.Dr. Erkan DUMAN

## CEVAPLAR





Adı Soyadı:

No:

PROGRAMLAMA DİLLERİ  
BİLGİSAYAR MÜHENDİSLİĞİ

Sınav süresi 90 dakikadır. Cep telefonlarınızı kapatınız. Sınav sorumlularının talimatlarına uyunuz. Sınav başlangıcından itibaren ilk 15 dakikada sınavı terk etmeyiniz. Soruların tamamı cevaplanacaktır.

1	2	3	4	5	Toplam

## SORULAR

1. (20p) Aşağıdaki program parçası static scope ve dynamic scope tekniği ile çalışan iki farklı dilde çalıştırılırsa ekran çıktısı ne olur? Sorunun altındaki boşluğa cevabınızı yazınız.

```

01  program main;
02      var x, y, z : integer;
03
04      procedure sub1;
05          var a, y : integer;
06          begin{sub1}
07              a = x+z+10;
08              y = 100;
09              x = a+20;
10          end{sub1}
11
12      procedure sub2;
13          var a, x : integer;
14
15      procedure sub3;
16          var a, z : integer;
17          begin{sub3}
18              a = y
19              x = a+10;
20              z = 500;
21              call sub1;
22              y = x-z;
23          end{sub3}
24
25      begin{sub2}
26          a = 5;
27          call sub3;
28          z = z+x+100;
29      end{sub2}
30
31  begin{main}
32      x = 77;
33      y = 77;
34      z = x+y;
35      call sub2;
36      print x,y,z
37  end{main}

```

Static Scope	Dynamic Scope

2. (30p) Bir programlama dilinde aşağıdaki gibi main içerisinde sub altprogramı çağrılmaktadır. Her bir

şıktaki hangi parametre geçiş yöntemini kullanacağınız verilmiştir. Ekrana yazılacak olan sayısal değerleri hesaplayıp şık altındaki boşluğa yazınız.

function sub (int a, int b, int c) begin a := b + c; b := c + 1; print a, b, c; end	a) a,b,c nin hepsi call-by-value ise;
function main begin int i := 5; int j := 10; int k := 15; sub(i, j, j + k); print i, j, k; end	b) a ve b call-by-reference, c ise call-by-value ise ;
	c) a ve b call-by-value-result, c ise call-by-value ise;
	d) a,b,c nin hepsi call-by-name ise;

3. (15p) a) 5 terminal, 7 nonterminal ve 9 kurala sahip bir CFG grameri açıklayarak yazınız.  
b) Bellek sızıntısı ve sallanan pointer kavramlarını bir örnek üzerinden açıklayınız.
4. (15p) Aşağıdaki CFG için First ve Follow kümelerini açıklayarak elde ediniz.
- $$S \rightarrow ABD$$
- $$A \rightarrow a \mid BSB$$
- $$B \rightarrow b \mid D$$
- $$D \rightarrow d \mid \epsilon$$
5. (20p) Aşağıdaki CFG için LR(0) parsing DFA diyagramını açıklayarak çiziniz.
- $$E \rightarrow x \mid +x?E \mid EE$$

Başarılar Dileriz.  
Doç.Dr. Mehmet KARAKÖSE  
Yrd.Doç.Dr. Erkan DUMAN

## CEVAPLAR





Adı Soyadı:

Numarası:

**Programlama Dilleri Genel Sınavı**

**Soru 1. (25 puan)** Aşağıdaki Lisp'te yazılmış program verilmiştir. ? (power-set '(1 2)) sorgusu yapıldığında aşağıdaki program çıkış olarak ne verir.

<pre>(defun power-set(set)   (if (null set) '()       (let ((psetOfRest (power-set (cdr set))))         (append          (mapcar #'(lambda (subset) (cons (car set) subset))                   psetOfRest)          psetOfRest ))))</pre>	<b>CEVAP 1:</b>  $((1\ 2)\ (1)\ (2)\ NIL)$
---	--

**Soru 2. (25 puan)** Aşağıda Prologda yazılmış program verilmiştir. Bu program ne iş yapar. Tek cümle ile belirtiniz?

<pre>power(N,0,1):- !. power(N,K,R):- K1 is K-1,power(N,K1,R1),R is R1*N.</pre>	<b>CEVAP 2:</b>  $N^K$
---	------------------------------

**Soru 3. (50 puan)** Aşağıdaki boşlukları doldurunuz? (Her boşluk 5 puan)

- Fonksiyonel Diller basit ve ortogonal dillerdir. Bu dillerde fonksiyonlar doğrudan parametrelere uygulanır. C, C++ ve Java'da olduğu gibi değişkenlerle ve atamalarla uğraşmaz.
- Programlama dillerinde 2 çeşit soyutlama sınıfı vardır.  
Process (işlem) soyutlama ve veri soyutlama.
- Bir cümle (sentence) herhangi bir alfabede karakterlerden oluşan bir stringdir
- Bir lexeme bir dilin en alt seviyedeki sentaktik(syntactic) birimidir (örn., \*, sum, begin)
- Bir simge (token) lexemelerin bir kategorisidir (örn., tanıttıcı(identifier))
- Bir dil (language) cümlelerden oluşan bir kümedir
- Bir grammer ancak ve ancak iki veya daha fazla farklı ayrıştırma ağacı(parse trees) olan bir cümlesel biçim üretiyorsa belirsizdir (ambiguous) denir.
- Lisp listeleri, tek-bağlı liste(single-linked lists) olarak saklanır.
- Lisp'te CDR komutu ilk elemanı gittikten sonra geriye kalan listeyi return eder. Bu fonksiyon yerine rest de kullanılabilir.

23-01-2012  
Başarılar.  
A. Bedri ÖZER



Soru 1) Aşağıdaki program parçası static scope ve dynamic scope tekniği ile çalışan iki farklı dilde çalıştırılırsa ekran çıktısı ne olur? Sorunun altındaki boşluğa cevabınızı yazınız (20 puan)

```

01 program main;
02     var x, y, z : integer;
03
04     procedure sub1;
05         var a, y : integer;
06         begin{sub1}
07             a = x+z+10;
08             y = 100;
09             x = a+20;
10         end{sub1}
11
12     procedure sub2;
13         var a, x : integer;
14
15         procedure sub3;
16             var a, z : integer;
17             begin{sub3}
18                 a = y
19                 x = a+10;
20                 z = 500;
21                 call sub1;
22                 y = x-z;
23             end{sub3}
24
25             begin{sub2}
26                 a = 5;
27                 call sub3;
28                 z = z+x+100;
29             end{sub2}
30
31 begin{main}
32     x = 77;
33     y = 77;
34     z = x+y;
35     call sub2;
36     print x,y,z
37 end{main}

```

Static Scope	Dynamic Scope
261 -413 341	77 117 871

**Soru 2)** Bir programlama dilinde aşağıdaki gibi main içerisinde sub altprogramı çağrılmaktadır. Her bir şıkta hangi parametre geçiş yöntemini kullanacağınız verilmiştir. Ekranaya yazılacak olan sayısal değerleri hesaplayıp şık altındaki boşluğa yazınız (30 puan).

```
function sub(int a, int b, int c)
begin
    a := b + c;
    b := c + 1;
    print a, b, c;
end
```

```
function main
begin
    int i := 5;
    int j := 10;
    int k := 15;
    sub(i, j, j + k);
    print i, j, k;
end
```

a) a,b,c nin hepsi call-by-value ise;

35 26 25  
5 10 15

b) a ve b call-by-reference, c ise call-by-value ise ;

35 26 25  
35 26 15

c) a ve b call-by-value-result, c ise call-by-value

35 26 25  
35 26 15

d) a,b,c nin hepsi call-by-name ise;

35 26 41  
35 26 15



Adı Soyadı:

Numara:

Sınav süresi 60 dakikadır. Cep telefonlarınızı kapatınız. Sınav sorumlularının talimatlarına uyunuz. Sınav başlangıcından itibaren ilk 15 dakikada sınavı terk etmeyiniz. Soruların tamamı cevaplanacaktır. Sınav süresince öğrenci kimlikleriniz masanızın üzerinde olacaktır.

1	2	3	4	5	6	Toplam

SORULAR

1. Kendisine parametre olarak gelen iki tane değişkenin içeriklerini yer değiştiren bir C veya C++ alt-programı yazınız. Alt program çalışmasını bitirip, çağıran programa döndüğünde yapılan değişiklik geçerli olmalıdır. Hem alt-programı hem de nasıl çağırılacağını kodlayınız. (\*Not: C veya C++ default parametre geçirme yöntemi değer ile çağırmadır.) (20 puan)

2. Eğer 5 bitlik bir mimaride işaretli sayılar ikiye tamlar yöntemi ile temsil ediliyorsa -15 ile +16 sayılarının toplama işleminin nasıl gerçekleştiğini gösteriniz. (15 puan)

3. break ve continue komutları arasında ne fark vardır? (15 puan)

4. Aşağıdaki boşlukları doldurunuz?

Bir bilgisayarın belleği ve işlemcisi arasındaki bağlantı hızı o bilgisayarın hızını belirler. Program komutları (instructions) çoğu kez yukarıda bahsedilen bağlantı hızından çok daha hızlı yürütülebilir; bağlantı hızı bu yüzden bir  darboğaz  sebep olur. (10 puan)

İmperatif dillerin temel özellikleri 3 tanedir;

- Değişkenler (variables).
- Atama ifadesi (5 puan)
- İterasyon (5 puan)

5. Aşağıdaki Lisp program parçasığı ne iş yapmaktadır. Tek bir cümle ile yazınız? (15 puan)

```
(defun bedri (x)
  (if (< x 0) (- x)
      x))
```

Mutlak Değer

6. Aşağıda verilen PROLOG programı ne işe yaramaktadır. Bir cümle ile kutucuğun içine yazınız? (15 puan)

```
start:- write('input a= '),read(A),
        write('input b= '),read(B),
        write('input c= '),read(C),
        A >= 0, B >= 0, C >= 0,
        A < B+C, B < C+A, C < A+B,
        write('Tamam uygun').
```

Yığın esnasında

1111  
1000  
1000  
1000

CEVAPLAR

③ Break komutu ilk switch sonuna gider, oradan default altına gelir. Sonra break komutu ile switch sonuna gider. Sonra continue komutu ile switch sonuna gider. Sonra kontrol kısmına gelir.

```
void swap2 (int *a, int *b) {
  int temp = *a;
  *a = *b;
  *b = temp;
}
```

swap2 (8c, 8d)

redans ile

Başarılar.

Doç. Dr. Bedri ÖZER

Yrd. Doç. Dr. Erkan DUMAN

Özellikle bölgesel veriler temel alınarak TRB1 bölgesinde olası bir vakıf üniversitesinin başarılı olabileme durumunun analizi yapılmıştır. Sonuç ve öneriler bölümünde ise mevcut durum ve bölgesel verilerin analizi ışığında, olası bir vakıf üniversitesi girişimini başarıya ulaştırabilecek hususlar tartışılmıştır.

## 2. YÜKSEKÖĞRETİM SEKTÖRÜ

### 2.1. Üniversite Kavramı ve Dünyada Üniversiteler

Üniversite, Eflatun ve Aristo'nun hiçbir politik ve dini baskı unsuru olmadan öğrencileri ile felsefi tartışma yarattıkları ortamdan esinlenerek kurulan evrensel ölçekte bağımsız ve tüzel kişiliğe sahip kurumlardır. Üniversiteler, amaçları evrensel bilgiyi öğretmek ve yaymak olan, felsefi tartışma ortamında akıl sürecini duygusal sürecin önüne alarak kişilerin olayları görerek ve tartışarak farkına varmasını sağlayan ortamlardır<sup>1</sup>.

Yapılan bazı araştırmalara göre dünya üzerinde 20.300'den fazla üniversite olduğu bildirilmektedir<sup>2</sup>. Bu üniversiteler arasında, örneğin 1088 yılından bu yana faaliyette olan Bologna Üniversitesi gibi eski üniversitelerin yanı sıra, henüz bir yaşını doldurmuş birçok üniversite de yer almaktadır.

### 2.2. Üniversite Türleri

Üniversitelerin sağladığı eğitim hizmetleri işlevsel açıdan "yarı kamusal hizmet" olarak tanımlanabilir. Yani, bu hizmetler "devlet veya diğer kamu tüzel kişilerin üretilmekle birlikte, tam kamusal nitelikte değildir. Bunlar, bölünebilir ve pazarlanabilir nitelikteki hizmetlerdir. Bu niteliklerinden dolayı fiyatlandırılabilirler" (Söyler, 2008: 57-58). Dolayısıyla, yükseköğretim hizmetleri devlet tarafından sağlanabildiği gibi, özel sektör tarafından da belirli fiyatlarla sağlanabilir. Bir ayırım yapılacak olursa, yükseköğretim hizmetlerinin devlet eliyle sağlandığı ve yürütüldüğü üniversiteler "kamu üniversitesi" veya "devlet üniversitesi", özel sektör eliyle sağlandığı kurumlar ise "özel üniversite" olarak adlandırılabilir.

### 2.3. Dünyada Özel Yükseköğretim Sektörünün Büyüklüğü

OECD tarafından 2001 yılından bu yana her yıl yayınlanmakta olan *Education at a Glance 2011: OECD Indicators* ( Bir Bakışta Eğitim 2011: OECD Göstergeleri) adlı raporun bulgularına göre, yükseköğretim daha iyi istihdam imkânlarına erişim ve daha iyi sosyal

<sup>1</sup> <http://tr.wikipedia.org/wiki/%C3%9Cniversite> (Erişim: 11.06.2012)

<sup>2</sup> <http://www.webometrics.info/> (Erişim: 11.06.2012)

SORU 1: C dilinde aşağıdaki fun isimli bir fonksiyon ve bu fonksiyonun kullanıldığı main programı tanımlanmıştır. a ve b şıklarındaki öncelik durumlarında sum1 ve sum2 değişkenlerinin değerlerini hesaplayınız. (25 puan)

a.) Soldan sağa doğru

sum1 =  
sum2 =

b.) Fonksiyon referansı/  
öncelikli

sum1 =  
sum2 =

```
int fun(int *k){  
    *k+=4;  
    return 3*(*k)-1;  
}
```

```
void main(){
```

```
    int i=10, j=20 sum1, sum2;
```

```
    sum1=(i/2)+fun(i);
```

```
    sum2=fun(j)+(j/2);  
}
```

SORU 2: Aşağıdaki program parçasına karşılık gelen döngü yapısını C, C++ veya Java dillerinden birisini kullanarak yazınız. (25 puan)

C, C++ veya Java dilindeki karşılığı:

$$for(k = (j+13)/27; k <= 10; k++)$$
$$\{$$
$$i = 3 * k - 1$$
$$\}$$

k:=(j+13)/27

loop:

if k>10 then goto out

k:=k+1

i:=3\*k-1

goto loop

out: ...

SORU 3: Aşağıda Lisp programlama dilinde yazılmış bir program verilmiştir. Bu programın ne iş yaptığını tek bir cümle ile söyleyiniz? (Cevabınızı yandaki boşluğa yazınız.)

(defun fonksiyon (n)	
(if (<= n 1)	
1	
(* n (fonksiyon (- n 1))))	

faktoriyel

SORU 4: Aşağıda Prolog programlama dilinde yazılmış bir program verilmiştir. Bu programın ne iş yaptığını tek bir cümle ile söyleyiniz? (Cevabınızı yandaki boşluğa yazınız.)

gs(1,1)	
gs(2,1)	
gs(N,R):-N>=3,N1 is N-1,N2 is N-2	
gs(N1,R1),gs(N2,R2),R is R1+R2	

Fibonacci

(n-1) + (n-2)

Not: Tüm cevaplar soru kâğıdı üzerine yapılacaktır. Cevap kâğıdı verilmeyecektir. Süre 40 dk.

Tablo 2.1. Kurum Tipine Göre Yükseköğretim Öğrenci Yüzdeleri (2009)

	Kurum Tipi					
	En az 2 Yünlük Okullar			3 yıl üzeri Okullar, YL ve Doktora dâhil		
	Kamu	Kamuya- bağımlı Özel	Bağımsız Özel	Kamu	Kamuya- bağımlı Özel	Bağımsız Özel
OECD	1	2	3	4	5	6
Avustralya	84,2	4,0	11,8	96,2	a	3,8
Avusturya	70,3	29,7	x(2)	84,8	15,2	x(5)
Belçika(1)	44,2	55,8	m	41,8	58,2	m
Şili	8,9	2,6	88,5	29,0	23,2	47,8
Cek Cum.	67,3	30,3	2,4	87,1	a	12,9
Danimarka	98,9	0,5	0,6	98,2	1,8	n
Estonya	46,6	16,9	36,5	0,2	91,2	8,6
Finlandiya	100,0	n	a	83,7	16,3	a
Fransa	70,0	8,4	21,6	85,0	0,8	14,2
Almanya (3)	57,5	42,5	x(2)	94,6	5,4	x(5)
Macaristan	54,2	45,8	a	86,4	13,6	a
İzlanda	30,5	69,5	n	79,5	20,5	n
İrlanda	97,6	a	2,4	96,6	a	3,4
İsrail	33,6	66,4	a	9,3	77,4	13,3
İtalya	87,2	a	12,8	92,4	a	7,6
Japonya	7,8	a	92,2	24,6	a	75,4
Kore	3,3	a	96,7	24,6	a	75,4
Meksika	95,5	a	4,5	65,9	a	34,1
Yeni Zelanda	59,4	30,8	9,8	96,5	2,6	0,9
Norveç	43,2	56,8	x(2)	85,8	14,2	x(5)
Polonya	74,9	a	25,1	66,6	a	33,4
Portekiz	97,0	a	3,0	75,7	a	24,3
Slovak Cum.	81,9	18,1	n	86,7	n	13,3
Slovenya	80,2	4,4	15,4	91,6	5,0	3,4
İspanya	79,7	14,5	5,7	89,7	n	10,3
İsveç	58,4	41,6	n	93,1	6,9	n
İsviçre	34,0	35,3	30,7	95,3	3,1	1,6
Türkiye	96,4	a	3,6	93,4	a	6,6
Birleşik Krallık	a	100,0	n	a	100,0	n
A.B.D.	79,1	a	20,9	71,5	a	28,5
OECD Ortalaması	61,6	20,7	17,7	70,7	14,0	15,3
EU21 Ortalaması	71,8	20,4	7,8	76,2	15,7	8,2
Diğer G20						
Arjantin (2)	58,7	17,1	24,2	79,8	a	20,2
Brezilya	15,0	a	85,0	27,5	a	72,5
Endonezya	47,9	a	52,1	38,3	a	61,7
Rusya Federasyonu (3)	95,2	a	4,8	83,1	a	16,9

Kaynak: Education at a Glance 2011: OECD Indicators, s.307'de bulunan tablodan kısaltılarak uyarlanmıştır.



ADI SOYADI:

Numarası:

Programlama Dilleri Dersi Bütünleme Sınavı

28-01-2013

SÖRÜ 1: C dilinde aşağıdaki gibi bir program parçacığı yazılmıştır. Kod çalıştıktan sonra j değişkeninin alacağı değeri hesaplayınız. (25 puan)

```
j=-3;
for(i=0; i<3; i++) {
    switch(i+2) {
        case 3:
        case 2: j--; break;
        case 0: j+=2; break;
        default: j=0;
    }
    if (i>0) break;
    j = j+1;
}
```

SÖRÜ 2: Aşağıdaki pascal program parçacığına karşılık gelen kodu if yapısı kullanarak C, C++ veya Java dillerinden birisinde yazınız. (25 puan)

=C, C++ veya Java dilindeki karşılığı:

```
Input = Index-1;
if (Input == 2 || Input == 4)
    even += 1;
else {
    if (Input == 1 || Input == 3)
        odd += 1;
    else
        if (Input == 0)
            zero += 1;
        else
            error = true;
}
```

```
case Index-1:
    2,4: even:=even+1;
    1,3: odd:=odd+1;
    0: zero:=zero+1;
else error:=true;
end
```

SÖRÜ 3: Aşağıda Lisp programlama dilinde yazılmış bir program verilmiştir. Bu programın ne iş yaptığını tek bir cümle ile söyleyiniz? (Cevabınızı alttaki boşluğa yazınız.)

```
(defun power-set(set)
  (if (null set) '({})
      (let ((psetOfRest (power-set (cdr
set))))
        (append
         (mapcar #'(lambda (subset) (cons (car
set) subset)) psetOfRest)
         psetOfRest
         )))
  )
```

All k3m bulma

SÖRÜ 4: Aşağıda Prolog programlama dilinde yazılmış bir program verilmiştir. Bu programın ne iş yaptığını tek bir cümle ile söyleyiniz? (Cevabınızı yandaki boşluğa yazınız.)

```
ToList(X,[Y|T],[X,Y|T]):- Y>=X.
ToList(X,[Y|T],[Y,X|T]):- X>Y.
insert([],[]).
insert([X],[X]).
insert([H|T],S):-
  insert(T,ST),
  ToList(H,ST,S).
```

Insertion Sort

Not: Tüm cevaplar soru kâğıdı üzerine yapılacaktır. Cevap kâğıdı verilmeyecektir. Süre 40 dk.

çıkımlar ile ilişkilendirilmektedir. Dolayısıyla bireylerin yükseköğretim programlarından mezuniyet oranları artmaktadır. Bu durum gelecekte yükseköğretime katılımın artan bir şekilde devam edeceğinin de bir göstergesi sayılabilir. Yükseköğretime katılım, günümüzün bilgiye dayalı toplumunda, işgücü piyasasının önem verdiği yüksek düzeydeki beceriler ve bilginin bir ülkenin nüfusu tarafından ne derecede elde edinildiğinin kısmi bir göstergesi olarak yorumlanmaktadır (OECD, 2011: 293). Ancak artan öğrenci sayılarının hepsini massedecek kamu üniversiteleri sistemleri her yerde mevcut olmayabilir. Bu gibi durumlarda, özel üniversiteler aracılığıyla artan talebin yükseköğretime katılımı sağlanabilir. Nitekim özel üniversiteler tüm dünyada yükseköğretim sektörünün önemli aktörleridir.

Tablo 2.1.'den de görülebileceği gibi, bazı OECD ülkelerinde yükseköğretim hizmetleri çok yüksek oranlarda özel sektör tarafından yürütülmektedir. 2005 yılı istatistiklerine göre dünya genelinde, özel yükseköğretim kurumlarında okuyan öğrencilerin oranı 1985'de yaklaşık % 18 iken, 2005 yılında bu oran % 30'a yaklaşmıştır (YÖK, 2007a: 5). Bu eğilimin devam ettiği varsayılırsa, 2025 yılında tüm dünyada yükseköğretim öğrencilerinin yaklaşık %40'ının özel yükseköğretim kurumlarında eğitim göreceği kestirimi yapılabilir.

Beşimleyici (Descripter) : Complete ile anında öğrenim başını ve sonunu hafızada tutulması için litanı vardır.

Complete

Complete



Adı Soyadı:

Numara:

Sınav süresi 45 dakikadır. Cep telefonlarınızı kapatınız. Sınav sorumlularının talimatlarına uyunuz. Sınav başlangıcından itibaren ilk 15 dakikada sınavı terk etmeyiniz. Soruların tamamı cevaplanacaktır. Sınav süresince öğrenci kimlikleriniz masanızın üzerinde olacaktır.

1	2	3	4	Toplam

SORULAR

1. Bir programlama dilini değerlendirirken hangi kriterlerden bahsetmek gerekir? C ve Java dillerini bu kriterlere göre kıyaslayınız. (25 puan)

2. Derleyici ile yorumlayıcı arasında ne fark vardır? Birbirlerine göre avantaj ve dezavantajlarını izah ediniz. (25 puan)?

3. Aşağıda PROLOG programının bir kısmı (gerçekler) verilmiştir. Notu 65'den yukarı olanlar başarılı sayılması için programın altına bir kural yazınız? (25 puan)

öğrenci (hakan, 68).  
öğrenci (seyhan, 81).  
öğrenci (gürhan, 40).  
öğrenci (nuran, 55).  
öğrenci (cikan, 95).

başarılıdır (İsim) :-  
öğrenci (İsim, not), not > 65

4. Aşağıdaki LISP programı ne iş yapmaktadır. Tek bir cümle ile belirtiniz? (25 puan)

```
(defparameter *small* 1)  
(defparameter *big* 100)  
  
(defun abcd ())  
  (setf *small* *big*) -1  
  
(defun smaller ())  
  (setf *big* (1- (abcd)))  
  (abcd))  
  
(defun bigger ())  
  (setf *small* (1+ (abcd)))  
  (abcd))  
  
(defun start-over ())  
  (defparameter *small* 1)  
  (defparameter *big* 100)  
  (abcd))
```

İlk 100 arasında sayı tahmin etme oyunu.

Başarılar.

Doç. Dr. Bedri ÖZER  
Yrd. Doç. Dr. Erkan DUMAN

CEVAPLAR

1. Okunabilirlik (Readability): Programın okunabilir ve anlaşılabilir kolylığı.
- Yazılabilirlik (Writeability): Bir dilin program yazmada kullanılabilir kolylığı. → Syntax, Sembolik.
- Güvenilirlik (Reliability): Sorulara uygunluk, soruları çözme. → Type checker, Stratejik, Farklı Algoritmalar.
- Maliyet (Cost): En az toplam maliyet.
- Ortogonalite → Elmede yapılar, kombinasyonlar birleştirip yeni yapılar oluşturulabilir.
- Tesirabilirlik, Genellik, İyi tanımlanmış olması.



SORU 1: Aşağıdaki PROLOG programının ne işe yaptığını solundaki boşluğa açıklayınız (25 puan)

Quicksort, left bir elemanı  
seçer listeyi bölür, pivot  
denilen bu elemanı kullanarak  
sağ ve sol listeleri oluşturur,  
sol'a küçükleri, sağ'a büyükleri  
Sıralanmış liste  
[ sol, pivot, sağ ] olur.

```
qs([X|Xs], Ys) :-  
    partition(Xs, X, Left, Right),  
    qs(Left, Ls),  
    qs(Right, Rs),  
    append(Ls, [X|Rs], Ys).  
qs([], []).  
  
partition([X|Xs], Y, [X|Ls], Rs) :-  
    X <= Y, partition(Xs, Y, Ls, Rs).  
partition([X|Xs], Y, Ls, [X|Rs]) :-  
    X > Y, partition(Xs, Y, Ls, Rs).  
partition([], Y, [], []).  
  
append([], Ys, Ys).  
append([X|Xs], Ys, [X|Zs]) :-  
    append(Xs, Ys, Zs).
```

SORU 2: Aşağıdaki LISP programının ne işe yaptığını solundaki boşluğa açıklayınız (25 puan)

Küçükten  
büyüğe  
Sıralama

```
(defun fun1 (l)  
  (if (endp l) nil  
      (let ((a (car l)) (r (cdr l)))  
        (append (fun1 (list-of (x in r) (< x a)))  
                  (list a)  
                  (fun1 (list-of (x in r) (>= x a)))))))  
  x < a  
  x >= a
```

SORU 3: identifer (<id>) kümesi={A,B,C} olan ve sadece toplama ile çarpma işlemlerini içeren atama kurallarından oluşan bir dilin gramerini yazınız. A = A + B \* C atama işlemini yazılan gramer kurallarından türetiniz. (Not: Çarpma işleminin öncelikli olmasını sağlayacaksınız. 30 puan)

SORU 4: Aşağıda terimlerin açıklamalarını altlarındaki birer satır boşluklarına yazınız. (20 puan)

- Scope** Bir değişkenin kapsamı, değişkenin yürütülebilir olduğu komutların içindeki alandır.
- Lifetime** Bir değişkenin belirli bir bellek hücreye bağlanması ile bu hafızanın kapatılması arasında geçen süredir.
- Type Checking** Programın çalışması veya derlenmesi sırasında verilen programın tip hatalarının test edilmesi işlemidir.
- Binding** Bir program elemanı arasında ilişki kurulması, birdey (başlama) denir. Başlatma zamanının statik ve dinamik olması ile çeşitliliği vardır.

