

Öğrenci Numarası : _____ Adı Soyadı : _____

Sınav ,çoktan seçmeli ve klasik olmak üzere iki kısımdan oluşmaktadır. Çoktan seçmeli kısmın cevapları optik okuyucuya uygun cevap kağıtları üzerine işaretlenecektir. İlk **30dk** sonunda optik cevap kağıtları toplanacaktır.
Kalan klasik soruları bu süreden sonra cevaplayabilirsiniz. Grubunuzu işaretlemeği unutmayınız!

Soru	1	2	3	4	5	6	Toplam
Puan	50	10	10	10	10	10	100
Not							

1. (1) (2P) _____ sabitler kalıp deyimlerin başka bir ismidir.(A _____ constant is a name for a literal.)
A. Statik(static) B. Tebliğ(manifest) C. Derleme zamanı(compile-time) D. Sembolik(symbolic)
- (2) (2P) Bir gramer _____ bağlamdan bağımsız olarak duşunu lu r(A grammar is considered to be context-free if _____.)
A.her bir nonterminal nerede yer aldığına bakmaksızın sağ tarafındaki herhangi bir seçenek ile yer değiştirebiliyorsa(each nonterminal can be replaced by any right-hand side choice, no matter where it appears) B. her bir terminal nerede yer aldığına bakmaksızın sağ tarafındaki herhangi bir seçenek ile yer değiştirebiliyorsa(each terminal can be replaced by any right-hand side choice, no matter where it appears) C. içerik metinleri gramer kurallarının sol kısmında yer alıyorsa(context strings appear on the left-hand sides of grammar rules) D. gramer aynı zamanda dilin anlamsallığını da ifade ediyorsa(the grammar also expresses the semantics of the language)
- (3) (2P) Bir prosedür çağırmacı ortamı ile _____ yoluyla haberleşir.(A procedure communicates with its calling environment through _____.)
A.yerel değişkenler(local variables) B. paylaşımlı bellek(shared memory) C. sabitler(constants) D. parametreler(parameters)
- (4) (2P) Yığıt çözme işlemi bir istisnai durum oluştuğunda ve kontrol _____ gerçeksir.(Stack unwinding occurs when an exception is thrown and control is _____.)
A.yığıtta başka bir konuma yerleştirildiğinde(placed elsewhere on the stack) B. bir işleyici ararken bir üstte çıktığında(exited back in search of a handler) C. yığıttan yığma belleğe geçtiğinde(moved off the stack and onto the heap) D. sürdürülemediğinde (unable to resume)
- (5) (2P) Bir fonksiyonda işlenenler(operands) _____ olarak görü lü r.(In a function, the operands are viewed as _____.)
A.işlemler(operators) B. argümanlar(arguments) C. global değişkenler(global variables) D. yerel değişkenler(local variables)
- (6) (2P) Aşağıdakilerden hangisi bir andı c(token) olarak duşunu lemez?(Which of the following is not considered to be a token category?)
A. boş alan(white space) B. kalıp deyim(literal) C. kimlik tanıtıcı(identifier) D. "özgü lenmi,ss"ozcu" k(reserved word)

- (7) (2P) _____a_samasında c,eviri programı ardışık karakterleri alır ve bunları andıc_lara(token) d'önu' ,stü'ru'r.(During its _____phase, a translator collects sequences of characters from the input program and forms them into tokens.)
A.u'retme(generating) B. ayrıştırma(parsing) C. tarama(scanning) D. analiz (analyzing)
- (8) (2P) _____atama işlemlerinde tip doğruluğu için kullanılan terimdir.(_____is a term used for the type correctness of assignments.)
A.Geriye d'önu'k uyumluluk(Backwards compatibility) B. Değer atama(Value assignment) C. Atama bağlama(Assignment linking) D. Atama uyumluluğu(Assignment compatibility)
- (9) (2P)_____olarak gönderme girişte parametrenin değerini kopyalar ve icra bittiğinde parametrenin son değerini geri kopyalar.(Pass _____copies in the parameter value, and at the end of execution, copies out the final value of the parameter.)
A.Adres(by address) B. Değer sonucu(by value-result) C. Referans(by reference) D. Değer(by value)
- (10) (2P) _____değerlendirme bir Bool ifadenin doğruluk değeri kesinleştiğinde sonlanır.(_____evaluation stops once the truth value of a Boolean expression is known.)
A.En iyilenmiş(Optimized) B. Uygun sıralı(Applicative order) C. Kısa devre(Short-circuit) D. Kestirimci(Predictive)
- (11) (2P)_____dildeki bu'tu'n programlar için bir ilk anlam verir fakat yeniden de tanımlanabilir. (_____have been given an initial meaning for all programs in the language but are capable of being redefined.)
AÖntanımlı kimlik tanıtıcılar(Predefined identifiers) B. Global kalıp deyim-ler(Global literals) C. Anahtar kelimeler(Keywords) D. Sabitler(Constants)
- (12) (2P) Eğer bilgi ,alışma zamanında kontrol ediliyorsa kontrol işlemine_____denir.(If information is maintained and checked at runtime, the checking is_____.)
A.,cıkırımsal(inferential) B. dinamik(dynamic) C. statik(static) D. ima yollu(referential)
- (13) (2P)_____tamsayılar gibi saklanabilen niceliklerdir.(_____are any storable quantities, such as integers.)
A. Veri(Data) B. Kimlik tanıtıcılar(Identifiers) C. Konumlar(Locations) D. Değerler(Values)
- (14) (2P)_____ortamda tüm bellek tahsisleri yükleme anında gerçekleştirilebilir ve tüm değişkenlerin konumları programın icrası süresince sabittir.(In a(n) environment, all memory allocation can be performed at load time, and the location of all variables are fixed for the duration of program execution.)
A.global B. evrensel(universal) C. tamamıyla statik(fully static) D. dinamik(dynamic)
- (15) (2P) Tip bildirimi_____sürecidir.(Type declaration is the process of_____.)
A. veri tiplerini kimlik tanıtıcılarla ilişkilendirme(associating data types to identifiers) B.tip kurucularını oluşturma(creating type constructors) C. yeni veri tiplerinin çıkarımı(inferring new data types) D. yeni veri tiplerini isimlendirme(naming new data types)

- (16) (2P) Statik tip sistemine sahip olmayan diller genellikle_diller olarak adlandırılır. (Languages without static type systems are usually called_languages.)
A. Güçlü tipli(strongly typed) B. zayıf tipli(weakly typed) C. ters tipli(reverse typed) D. tipsiz(untyped)
- (17) (2P) Postfix ve prefix biçimindeki ifadelerin avantajı _____.(The advantage of postfix and prefix forms for expressions is that_.)
A. hangi işlemin hangi sırada yapılacağına karar vermek için parantezlere ihtiyaç duymamasıdır.(they do not require parentheses to express the order in which operators are applied) B. İşlemlerin birleşmesi için katı kurallar uygulanmasıdır.(they follow strict rules that define associativity of operators) C. tekli işlemlere izin vermemeleridir.(they do not allow unary operators) D. işlem "önceliği için katı kurallar izlemesidir(they follow strict rules of operator precedence)
- (18) (2P) _____ bir prosedür bloğundaki yerel değişkenler için ayrılan bellek alanına denir.(The _____ is the memory allocated for the local objects of a procedure block.)
A. yığılma bellek kaydı(heap record) B. aktivasyon kaydı(activation record)
C. çağrı kaydı(call record) D. aktivasyon yığılma belleği(activation heap)
- (19) (2P) Tipleri ifadelere tutturma süreci tip_____olarak adlandırılır.(The process of attaching types to expressions is called type_.)
A. çıkarımı(inference) B. kontrolü(checking) C. oluşturma(construction)
D. eşdeğerliği(equivalence)
- (20) (2P) _____ bir kuralın kısımlarını birbirinden ayırt etmek için kullanılan sembollerdir.(_____ are symbols used to distinguish between parts of a rule.)
A. Meta semboller(Metasymsbols) B. Kimlik tanıttıcılar(Identifiers) C. Ayırt ediciler(Distinguishers) D. Soyut karakterler(Abstract characters)
- (21) (2P)_____ mevcut aktivasyon kaydının konumunu saklar.(The_____ maintains the location of the current activation record.)
A. Yığıt işaretçisi(stack pointer) B. Ortam işaretçisi(environment pointer)
C. Aktivasyon işaretçisi(activation pointer) D. Yığıt kaydedicisi(stack register)
- (22) (2P) Bir değişkenin değerindeki değişimlik bir komutun icrasının ötesinde etkili oluyorsa buna _____denir.(When a change in the value of a variable persists beyond the execution of a statement, a(n)_has occurred.)
A. yan etki(side effect) B. bileşik değişimlik(compound change) C. hata(error)
D. kalıcı etki(persistent effect)
- (23) (2P) Paylaşarak atama _____yoluyla yapılır.(Assignment by sharing works by _____.)
A. bellek adreslerinin kopyasını çıkartma(duplicating memory addresses) B. paylaşımli belleğe erişme(accessing shared memory) C. değeri yeni bir konuma kopyalayama(copying the value to a new location) D. ilk değişkenin konumunu ikinci değişkene kopyalama(copying the location of the first variable to the second)
- (24) (2P) Java ve Smalltalk gibi nesneye dayalı programlama dillerinde_____ başka bir sınıf bildirimi içinde yer alması gerekmeyen tek bildirimdir.(In object-oriented languages such as Java and Smalltalk, the _____ is the only declaration that does not itself need to be inside another class declaration.)
A. dosya(file) B. paket(package) C. modül(module) D. sınıf(class)

(25) (2P) Eğer kontrol istisnai durumun ilk ortaya çıktığı konuma tekrar dönüyorsa _____ modeli kullanılmıştır.(If control resumes at the point where the exception was first raised, the _____ model is being used.)

- A. su"rdü"rme(resumption) B. kefareet(redemption) C. sonlandırma(termination)
D. kurtarma(recovery)

2. (10P) Parametre olarak iki liste alan ve bu listelerin elemanlarını tek bir listede birleştiren `birlestir` Haskell fonksiyonunu yazınız. Fonksiyonun "örnek kullanımı" aşağıda verilmiştir.(Write `birlestir` Haskell function that takes 2 lists as parameters and combines each item on lists to a single list and returns it. Sample usage of the function is given below.)

```
*Main> birlestir [1,2,3,4] [5,6]
[1,5,2,6,3,4]
*Main> birlestir [1,2] [3,4,5,6,7]
[1,3,2,4,5,6,7]
*Main> birlestir "ab" "cdefg"
"acbdefg"
```

```
birlestir x [] = x
birlestir [] y = y
birlestir (x:xs) (y:ys) = x:y:(birlestir xs ys)
```

3. (10P) Bir listedeki negatif sayıları silen `del_negative` fonksiyonunu Prolog dilinde yazınız. `del_negative` fonksiyonunun "örnek kullanımı" aşağıda verilmiştir.(Write the `del_negative` function that removes negative elements from the list(first parameter) and puts the rest to the second list(second parameter). Sample usage of the function is given below.)

```
del_negative(Liste,YeniListe).
?- del_negative([-3, 4, 5, 6, -2, 7, 6, -10], Y).
Y = [4, 5, 6, 7, 6] .
```

```
del_negative([],[]).
del_negative([B|K], YL) :- B<0, del_negative(K, YL).
del_negative([B|K], [B|YL]) :- B>=0, del_negative(K, YL).
```

4. (10P) Elimizde `int x[10]; int y[10];` dizilerinin olduğunu düşünelim. `x = y` atama işlemi C dilinde neden derlenmez? Atama işlemi çalışacak şekilde bildirimler düzenlenebilir mi? Gösteriniz. (Suppose that we have two C arrays: `int x[10]; int y[10];` Why won't the assignment `x = y` compile in C? Can the declarations be fixed so the assignment will work?)

Atama işlemi derlenmez çünkü C dizileri sabit olan ve yeniden değer atanamayan işaretçiler olarak değerlendirir. İşaretçinin gösterdiği konum yığıtta otomatik olarak tahsis edilir. Bildirimleri düzeltmenin bir çok yolu vardır. Bunlardan birisi aşağıdaki gibidir.

```
int *x;
int y[10];
x = y; /* artık derlenir */
x[2] = 17; /* x dizi olarak değerlendirilebilir */
```

Bu durumda başka bir değeri atanana kadar `x`, `y`'nin takma adı olur(alias).

5. Aşağıdaki C fonksiyonu ile `VE(&&)` işleminin kısa devre değerlendirme sürümünü yazılmak istenmektedir. (Suppose that we were to try to write a short-circuit version of `and` in C as the following function:)

```
int and(int a, int b){
    return a ? b : 0;
}
```

- (a) (5P) Bu fonksiyon neden beklediği gibi çalışmaz? `and(0, 1 / 0);` ile `0 && 1/0` işlemlerini düşünerek farkı belirtiniz. (Why doesn't this work? State the difference of `and(0, 1 / 0);` and `0 && 1/0`)

Bu işlem çalışmaz çünkü C dilinde parametreler fonksiyona gönderilmeden önce değerlendirilir. Bu nedenle kısa devre değerlendirme yapılamaz ve `and(0, 1 / 0);` hata verir.

- (b) (5P) Normal sıralı değerlendirme kullanılsaydı çalışır mıydı? Açıklayınız. (Would it work if normal order evaluation were used? Why?)

Normal sıralı değerlendirme kullanılsaydı fonksiyon beklediği şekilde çalışırdı. `a` ve `b` ifadelerinin değerlendirilmesi gecikmeli olarak gerçekleşirdi ve `b` ifadesinin değerlendirilmesi için `a` ifadesinin değeri sıfırdan farklı olması gerekirdi.

6. Aşağıdaki C söz dizimine uygun olarak yazılan programın çıktılarını belirtilen parametre gönderme yöntemlerine göre yazınız. (Give the output of the following program (written in C syntax) using the four parameter-passing methods below:)

```
1 #include <stdio.h>
2 int i, a[2];
3 void p(int x, int y) {
4     x++;
5     i++;
6     y++;
7 }
8 void main() {
9     a[0] = 1; a[1] = 1;
10    i = 0;
11    p(a[i], a[i]);
12    printf("%d %d", a[0], a[1]);
13 }
```

(a) (2P) Değer olarak gönderme (pass by value)

1 1

(b) (2P) Referans olarak gönderme (pass by reference)

3 1

(c) (3P) Değer-sonucu olarak gönderme (pass by value-result)

2 1

(d) (3P) İsim olarak gönderme (pass by name)

2 2