106)Uzman sistem nedir. Nicin kullanilir.

111)Uzman sistemde asagidaki bilesenler mevcuttur.

1)
2)
..
..

107) **Uzman Sistemin Faydaları ndan onemilerini yaziniz.**

108)Uzman Sistemin Sınırları, cozemedigi problemler nelerdir

109)Uzman Sistemlerin Pratik Hayattaki Uygulamalarından orenkeler verin

112)Bir uzman sistemde bulunmasi gerekn temel bilesenlerden olan, Bilgi kazanma' yi aciklayin

112)Bir uzman sistemde bulunmasi gerekn temel bilesenlerden olan, Çıkarım mekanizması' ni aciklayin

*113)Bir uzman sistemde bulunmasi gerekn temel bilesenlerden olan, Düşünme kapasitesini iyileştirme' yi a*ciklayin

---------- ----------

521)
kadin(ayse). %ayse kadindir
kadin(fatmaanne).
kadin(elifteyze).
erkek(ahmet). %ahmet erkekdir.
erkek(bahribaba).
erkek(bilalamca).
erkek(cezmidede).
aile(bahribaba,cezmidede). %cezmidede bahribabanin ailesidir.
aile(bilalamca,cezmidede).
aile(ali,bahribaba).
aile(ahmet,bahribaba).
aile(ahmet,fatmaanne).
aile(ayse,bahribaba).
aile(ayse,fatmaanne)

521)Verilenleri kullanarak,
anne, baba, dede, teyze, hala tanimlari yapin.

531)
```
kural33 (A):-  A>10,  write(' A buyuk 10'),  nl.
kural33(A):-  A>9,  write(' A buyuk 9'),  nl.
kural33(A):-  A>8,  write(' A buyuk 8'),  nl.
```

```
?-kural33(1).
    Bilgisayar ekranina hangi yazi gelir.
?-kural33(9).
    Bilgisayar ekranina hangi yazi gelir.
?-kural33(11).
    Bilgisayar ekranina hangi yazi gelir.
```

532)
```
kural34(A):-  A>8,  write(' A buyuk 8'),  nl.
kural34(A):-  A>9,  write(' A buyuk 9'),  nl.
kural34 (A):-  A>10,  write(' A buyuk 10'),  nl.
```

```
?-kural34(1).
    Bilgisayar ekranina hangi yazi gelir.
?-kural34(9).
    Bilgisayar ekranina hangi yazi gelir.
?-kural34(11).
    Bilgisayar ekranina hangi yazi gelir.
?-kural34(1,11).
    Bilgisayar ekranina hangi yazi gelir.
?-kural34(11,1).
    Bilgisayar ekranina hangi yazi gelir.
```

541)
```
kuralq41(A):-  A>10,  write(' A buyuk 10'),  nl, fail.
kuralq41(A):-  A>20,  write(' A buyuk 20'),  nl.
kuralq41(A):-  A>30,  write(' A buyuk 30'),  nl.
?- kuralq41(5).
    Bilgisayar ekranina hangi yazi gelir.
?- kuralq41(15).
    Bilgisayar ekranina hangi yazi gelir.
?- kuralq41(55).
    Bilgisayar ekranina hangi yazi gelir.
```

542,543,544,545,.....600,601,.....
Yukaridaki sorulara benzer sorular.

ccccc

---------- ----------

problem 611)

class(gun, saat ,oda, ders numarasi ,ders_lab_uygulam).

sinif (pazartesi,10,b1,505201,teorik).

class(monday, 10, 343, 2.7, lecture).
class(monday, 11, 343, 2.7, tutorial).
class(monday, 11, 344, 1.9, lecture).
class(monday, 12, 343, 1.5, lecture).
class(monday, 14, 145, 1.1, lecture).
class(monday, 15, 343, 1.7, lecture).
class(monday, 16, 343, 1.1, tutorial).
class(tuesday, 9, 343, 1.9, lecture).
class(tuesday, 10, 343, 1.9, tutorial).
class(tuesday, 10, 145, 1.7, lecture).
class(tuesday, 11, 343, 1.3, lecture).
class(tuesday, 11, 145, 1.7, tutorial).
class(tuesday, 12, 343, 1.3, tutorial).
class(tuesday, 14, 343, 1.6, lecture).
class(tuesday, 15, 343, 1.6, tutorial).
class(tuesday, 16, 343, 2.3, lecture).
class(wednesday, 9, 343, 1.4, lecture).
class(wednesday, 9, 223, 1.1, laboratory).
class(wednesday, 9, 224, 1.5, laboratory).
class(wednesday, 10, 223, 1.1, laboratory).
class(wednesday, 10, 224, 1.9, laboratory).
class(wednesday, 11, 223, 2.7, laboratory).
class(wednesday, 11, 224, 1.9, laboratory).
class(thursday, 10, 343, 2.3, lecture).
class(thursday, 11, 145, 1.3, lecture).
class(thursday, 11, 343, 2.3, tutorial).
class(thursday, 14, 343, 1.8, lecture).
class(thursday, 15, 343, 1.8, tutorial).
class(friday, 9, 343, 1.3, lecture).
class(friday, 10, 343, 1.6, lecture).
class(friday, 11, 145, 1.5, lecture).
class(friday, 11, 343, 1.4, lecture).
class(friday, 12, 145, 1.5, tutorial).
class(friday, 12, 343, 1.4, tutorial).
class(friday, 14, 343, 2.7, lecture).
class(friday, 15, 145, 1.1, lecture).
class(friday, 15, 343, 1.8, lecture).

ders(505205,mikroislemci, mn).

course(1.1, functional_programming, jd).
course(1.1, functional_programming, ajf).

course(1.3, theory_of_functions, clh).
course(1.4, deductive_databases, mjs).
course(1.5, software_development, rjc).
course(1.6, artificial_intelligence, rak).
course(1.6, artificial_intelligence, frk).
course(1.6, artificial_intelligence, mjs).
course(1.7, concurrency, jk).
course(1.8, logic_programming, cjh).
course(1.9, prolog_problem_solving, frk).
course(2.3, semantics_of_programming_languages, clh).
course(2.7, natural_language_processing, drb).
course(2.7, natural_language_processing, cdsm).

hoca(mn,mehmet,akif).

lecturer(jd, john, darlington).
lecturer(ajf, tony, field).
lecturer(kb, krysia, broda).
lecturer(clh, chris, hankin).
lecturer(mjs, marek, sergot).
lecturer(rjc, jim, cunningham).
lecturer(rak, bob, kowalski).
lecturer(frk, frank, kriwaczek).
lecturer(jk, jeff, kramer).
lecturer(cjh, chris, hogger).
lecturer(drb, derek, brough).
lecturer(cdsm, chris, moss).

Asagidaki bilgileri ogrenmek icin gerekli prolog komutlarini yazin.

611)a. Is course 1.1 taught on Mondays?
1.1 nolu ders pazaresi (monday) varmi

612). Does lecturer MJS teach course 1.3?
MJS isimli hoca 1.3 nolu derse giriyormu

613). Does course 2.2 have any classes?
2.2 nolu dersin sinifi varmi.

614) Does course 1.1 have any classes in room 145?
1.1 nolu ders 145 nolu sinfta oldugu oluyormu.

615) Does course 1.6 have any laboratory sessions?

f. Is David the first name of lecturer DRB?
g. Are any laboratory sessions held in room 343?
h. Does Jeff Kramer teach any course?
i. On what day(s) is course 1.1 taught?

j. What course(s) are taught at 11 am on Mondays?
k. What course(s) does lecturer CLH teach?
622). On what days and at what times are the classes for course 2.3?

NB. You may need to use Prolog's built-in binary predicate \==
where T_1 \== T_2 *means that: term* T_1 *is NOT identical to term* T_2.

2. Compose Prolog clauses defining the following predicates:

a teacher(X) meaning *X is a lecturer.* teacher(X):- lecturer(X,_,_).
b. teaches(X) meaning *X teaches a course.* teaches(X):- course(_,_,X).
c. polymath(X) meaning *lecturer X teaches several courses.*

```
polymath(X) :- course(C1,_,X), course(C2,_,X), C1 \== C2.
```

d. overworked(X) meaning *lecturer X is either a polymath or teaches a 9 am course.*
e. clashes(X,Y) meaning *courses X and Y overlap in at least one period.*
f. miraculous(X) meaning *lecturer X teaches courses which clash.*

Assume that lecturers are referred to by their initials and courses by their numbers.

631) Which courses have classes in room 145?
632) Which courses have laboratory sessions?
633) What are the names of the courses taught by Marek Sergot?
634) Which courses have lectures and tutorials on Mondays?
635) Which courses have some classes in rooms other than 343?
636) Which courses, taught by a Chris, have a class on Thursdays?
s. Which courses have more than one lecturer?
t. Which lecturers share the same first name?
u. What are the surnames of the lecturers who teach more than one course?

---

711)

```
even(0).
even(X) :- X > 1, X1 is X-2, even(X1).
```

Yukaridaki proramin calisma prensibini asagida verilen ornekler uzerinde aciklayin.

?- even(0). yazilirsa ne olur.
?- even(1). yazilirsa ne olur.
?- even(7). yazilirsa ne olur.

---

713)
Problem 713: odd(X). programini yazin.
Problem 714: uce_bolunme(X). programini yazin.
Problem 715: altiya_bolunme(X). programini yazin. (3 e ve 2 ye bolunme kuralini kullanarak)

Problem 716: yediye_bolunme(X). programini yazin.

---

721)

```
aabb(0,1).
aabb(N,F) :-   N>0,     N1 is N-1,     aabb(N1,F1),     F is N * F1.
```

yukaridaki programa asagidaki sayilar yazilsa sonuc ne olur. neden. Aciklayin.

```
?-aabb(0,1).
?-aabb(0,2).
?-aabb(1,0).
?-aabb(0,XX).
?-aabb(XX,0).
```

---

722)

Sagdan soldan koselerden sayilarin toplami 15 olacak sekilde sayilari yerlestiren prolog programi yazin. her sayi farkli olacak.

2	9	4
7	5	3
6	1	8

723)

3x3 luk stranc tahtasinda her satirda ve her sutunda ve her kosegende sadece bir tane 1 olacak digerlerri sifir olacak bir program yazin.

1	0	0
0	0	1
0	1	0

724)Asagidaki program ne is yapar.

```
wws( [ ] , 0 ).
wws([ _ | L ] ,N) :- wws(L,N1), N is N1 + 1.
```

725) Asagidaki program ne is yapar.

```
 aaq( [ ] ,L,L).
aaq ( [ X | L ] , M , [ X | N ] ) :- aaq(L,M,N).
```

?- aaq([ ], [ ],)

726)

```
err(X,[X|_]).
err(X,[_|L]) :- err(X,L).

?-err(a,[a,b,c]).
?-err([a],[a,b,c]).
?-err([a,b],[a,b,c]).
?-err([a,b],[[a,b],c]).
```

727)

```
exx(X,[X|L],L).
exx(X,[Y|L],[Y|M]) :- exx(X,L,M).

?- exx(a,[a,b,c],X).
   X=?
?- exx(b,[a,b,c],X).
  X=?
?- exx(_,[a,b,c],X).
 X=?
```

727)

```
aax(X, Y, X) :- Y < X, !.
aax(X, Y, Y) :- Y >= X.
?-aax(3,4,X).
    sonucu nedir.
?-aax(11,5,X).
    sonucu nedir.
?-aax(3,4,5).
    sonucu nedir.
?-aax(11,5,11).
    sonucu nedir.
?-aax(11,11,11).
    sonucu nedir.
 ?-aax(11,10,11).
    sonucu nedir.
?-aax(11,12,11).
    sonucu nedir.
```