



Assignment 2

CSI2120 Programming Paradigms

Winter 2020

Due on March 18th before 11:00 pm in Virtual Campus

6 marks

There are [15 points] in this assignment. The assignment is worth 6% of your final mark.

All code must be submitted in prolog files. Screenshots, files in a format of a word editor, pdfs, handwritten solutions, etc. will not be marked and receive an automatic 0.

Reminder: Late assignments are not accepted.

Question 1. Prolog Search Tree [3 points]

Consider the following database:

```
weekday(monday) .  
weekday(tuesday) .  
weekday(wednesday) .  
weekday(thursday) .  
weekday(friday) .  
weekend(saturday) .  
weekend(sunday) .  
holiday(friday) .  
weather(monday,sunny) .  
weather(tuesday,snow) .  
weather(wednesday,sunny) .  
weather(thursday,cold) .  
weather(friday,snow) .  
weather(saturday,rain) .  
weather(sunday,cold) .  
ski(J):- weekend(J), weather(J,W),W\=rain.  
ski(J):- holiday(J), weather(J,snow) .  
ski(J):-weekday(J), weather(J,sunny),\+nicedayoff(J) .  
nicedayoff(J):- weekend(J), weather(J,sunny) .  
nicedayoff(J):- holiday(J), weather(J,sunny) .
```

- a) Draw the Prolog search tree for the following query:

```
?- ski(J) .
```

- b) What will be the effect of inserting a cut in the third clause of the predicate `ski/1`? Show clearly the impact of this cut on the Prolog search tree and on the solutions found.

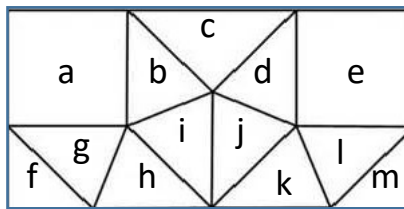
```
ski(J) :- weekday(J), weather(J, sunny), !, \+nicedayoff(J) .
```

- c) If instead this cut is inserted in the second clause of the predicate `ski/1`, what will be the effect?

```
ski(J) :- holiday(J), !, weather(J, snow) .
```

Question 2. Colouring [6 points]

The panes of a stained glass window are shown in the figure below.



The predicate `adj/2` encodes if two windowpanes are adjacent.

```
adj(a,b) .
adj(a,g) .
adj(b,c) .
adj(b,i) .
adj(c,d) .
adj(d,e) .
adj(d,j) .
adj(e,l) .
adj(f,g) .
adj(g,h) .
adj(h,i) .
adj(i,j) .
adj(j,k) .
adj(k,l) .
```

```
adj(l,m).
```

- a) The predicate `colorset/2` is to generate all sets of possible colours for a list of windowpanes.

```
color(red).
color(yellow).
color(blue).

?- colorset([b,c,d,i,j],C).
C = [red, red, red, red, red] ;
C = [red, red, red, red, yellow] ;
C = [red, red, red, red, blue] ;
C = [red, red, red, yellow, red] ;
C = [red, red, red, yellow, yellow] ;
C = [red, red, red, yellow, blue] ;
C = [red, red, red, blue, red] ;
C = [red, red, red, blue, yellow] ;
C = [red, red, red, blue, blue] ;
C = [red, red, yellow, red, red]
... (more solutions)
```

Define the predicate `colorset/2`.

- b) We want to create a stained glass window for which the adjacent windowpanes are of different colours. The `diffadjcolor/4` predicate is true if the list of windowpane colours are such that all neighboring windowpanes have different colours.

```
?- diffadjcolor(b,red,[c,d,i,j],[blue, red, blue, yellow]).
true.
?- diffadjcolor(b,blue,[c,d,i,j],[blue, red, blue, yellow]).
false.
```

Define the predicate `diffadjcolor/4`.

- c) A window is said to be valid if no two adjacent windowpanes have the same colour. The generator `generate/2` is to produce all valid combinations for a window.

```
generate(Gs,Cs):-colorset(Gs,Cs),valid(Gs,Cs).
```

Define the predicate `valid/2`.

```
?- generate([b,c,d,i,j],C).  
C = [red, yellow, red, yellow, blue] ;  
C = [red, yellow, red, blue, yellow] ;  
C = [red, yellow, blue, yellow, red] ;  
C = [red, yellow, blue, blue, red] ;  
C = [red, yellow, blue, blue, yellow] ;  
C = [red, blue, red, yellow, blue] ;  
C = [red, blue, red, blue, yellow] ;  
C = [red, blue, yellow, yellow, red] ;  
C = [red, blue, yellow, yellow, blue] ;  
C = [red, blue, yellow, blue, red] ;  
C = [yellow, red, yellow, red, blue] ;  
C = [yellow, red, yellow, blue, red] ;  
C = [yellow, red, blue, red, yellow] ;  
C = [yellow, red, blue, blue, red] ;  
C = [yellow, red, blue, blue, yellow] ;  
C = [yellow, blue, red, red, yellow] ;  
C = [yellow, blue, red, red, blue] ;  
C = [yellow, blue, red, blue, yellow] ;  
C = [yellow, blue, yellow, red, blue] ;  
C = [yellow, blue, yellow, blue, red] ;  
C = [blue, red, yellow, red, blue] ;  
C = [blue, red, yellow, yellow, red] ;  
C = [blue, red, yellow, yellow, blue] ;  
C = [blue, red, blue, red, yellow] ;  
C = [blue, red, blue, yellow, red] ;  
C = [blue, yellow, red, red, yellow] ;  
C = [blue, yellow, red, red, blue] ;  
C = [blue, yellow, red, yellow, blue] ;  
C = [blue, yellow, blue, red, yellow] ;  
C = [blue, yellow, blue, yellow, red] ;  
false.
```

Question 3. List processing [2 points]

Five friends want to go on a trip. They must choose which destination they will visit. To do this, each friend list three countries in order of preference. i.e.,

```
choice(marie, [peru,greece,vietnam]).
choice(jean,  [greece,peru,vietnam]).
choice(sasha, [vietnam,peru,greece]).
choice(helena,[peru,vietnam,greece]).
choice(emma,  [greece,peru,vietnam]).
```

The country will be chosen according to the following formula: a first choice is worth 3 points, a second choice is worth 2 points and the last choice is worth 1 point. The country to visit will be the one collecting the most points.

Write the predicate `where/2` that performs this calculation.

```
?- where([marie,jean,sasha,helena,emma],Country).
    peru .
```

Question 4. Looping [4 points]

A list of integers is to be generated with numbers that are not divisible by any of a specified set of numbers.

- a) Design a predicate `divisible/2` that is true if a given number can be divided by any of the numbers in the list.

```
?- divisible([5,7], 15 ).  
    true .
```

```
?- divisible([5,7], 9 ).  
    false .
```

- b) Design a predicate `generateList/3` that returns a list of positive integers of length `N` such that none of the integers is divisible by any of the numbers in the List `D` and at the same time is smaller equal than its rank. Do not include 0 in the list.

```
? - generateList([3,5],5,L) .  
[ 1, 1, 1, 1, 1] ;  
[ 1, 2, 1, 1, 1] ;  
[ 1, 1, 2, 1, 1] ;  
[ 1, 1, 1, 2, 1] ;  
[ 1, 1, 1, 1, 2] ;  
[ 1, 2, 2, 1, 1] ;  
... (more solutions)
```