(Attempt 2 questions out of 3)

Question 1.

(a) Suppose a Prolog knowledge base contains the definition of a predicate `size` such that `size(P, S)` means that person `P` is `S` centimetre tall. Formulate queries that correspond to the following questions:

   (i) Is John 192 cm tall?

  (ii) Is Peter as tall as Edgar?

 (iii) Who has the same size as John (and is different from John)?

 (iv) Is Julie smaller than Sue?

  (v) Who is taller than Lisa?

 (vi) Is it true that nobody is 200 cm tall?

[10 Marks]

(b) Consider the two programs `P1` and `P2`:

```
%P1                                    %P2

p(a).              %1                  p(Z) :- p(Z).       %1
p(Z) :- p(Z).      %2                  p(a).               %2
```

Determine the answers to the query `?- p(X).` that Prolog will give when repeatedly pressing `;`, by drawing adequate parts of the derivation trees for the two programs.

[6 Marks]

(c) In the course we have discussed two possible search strategies which can be used in general to find successful nodes in derivation trees. Describe the two search strategies (you may illustrate them by drawing some adequate trees). Which one is usually implemented in Prolog? Discuss advantages and disadvantages of the two search strategies.

What would be the answers to the queries in (b) when using the alternative search strategy?

[9 Marks]

Question 2.

(a) Say for the following queries whether they will fail or succeed or cause an error. If the query fails or causes an error, explain why this is the case. If the query succeeds, give the answers that Prolog will return.

  (i) `?- X =  1 + Y.`

  (ii) `?- X == 1 + Y.`

  (iii) `?- X is 1 + Y,  Y is 2.`

  (iv) `?- Y is 2,  X is 1 + Y.`

  (v) `?- [X] = [X,X].`

  (vi) `?- [X] = [X|X].`

[10 Marks]

(b) Consider the program `SUM` defining the ternary predicate `sum` by

```
sum(0,Y,Y).                          %1
sum(s(X),Y,s(Z)) :- sum(X,Y,Z).    %2
```

Extend the following SLD-resolution step of the query `?- sum(s(s(0)),s(0),R).`

$$\texttt{sum(s(s(0)),s(0),R)} \overset{C_1,\theta_1}{\Longrightarrow} \texttt{sum(s(0),s(0),Z1)}$$

variant C_1 of `%2`: `sum(s(X1),Y1,s(Z1)) :- sum(X1,Y1,Z1).`
unifier θ_1: `{ X1/s(0), Y1/s(0), R/s(Z1) }`

to a successful SLD-derivation, and precisely state all variants of clauses of the program `SUM` and the most general unifiers used within this derivation. Extract the computed answer substitution by first computing the composition of all unifiers obtained in this derivation.

[8 Marks]

(c) Which of the following terms are unifiable? In case the terms are unifiable, compute their most general unifier, otherwise explain why they are not unifiable.

  (i) `g(X,f(X,Y),X)`   `g(f(a,U),Z,f(U,U))`

  (ii) `g(X,f(X,Y),X)`   `g(f(a,Z),U,f(U,U))`

[7 Marks]

Question 3.

(a)　(i) Explain the general effect of the cut predicate on Prolog's search trees.

　　(ii) Explain how the cut is used to define *if-then-else* in Prolog. How can this be used to define *negation as failure*?

[8 Marks]

(b) Consider the program

```
p(a).
p(b) :- !.
p(c).
```

Draw the resolution trees for the following three queries, and for each query predict the subsequent answers that Prolog will return.

　(i) `?- p(X).`

　(ii) `?- X=c, p(X).`

　(iii) `?- p(X), X=c.`

[9 Marks]

(c) Write a program that defines a binary predicate `double` such that for lists `L` and `M`, `double(L,M)` holds if and only if `M` is obtained by repeating each element in `L`. For example, `double([a],[a,a])` and `double([a,b,c],[a,a,b,b,c,c])` are true, and `double([a,b],[a,b,a,b])` and `double([a,b],[a,a,a,b,b])` are false.

To ensure your program is correct, draw the resolution trees for the given example queries.

[8 Marks]