

Practical 4

1. Write a 3-place predicate `combine1` which takes three lists as arguments and combines the elements of the first two lists into the third as follows:

```
?- combine1([a,b,c],[1,2,3],X).
```

```
X = [a,1,b,2,c,3]
```

```
?- combine1([foo,bar,yip,yup],[glub,glab,glib,glob],Result).
```

```
Result = [foo,glub,bar,glab,yip,glib,yup,glob]
```

```
combine1([],[],[]).
```

```
combine1([X|Xs],[Y|Ys],[X,Y|Zs]) :- combine1(Xs,Ys,Zs).
```

2. Write a 3-place predicate `combine2` which takes three lists as arguments and combines the elements of the first two lists into the third as follows:

```
?- combine2([a,b,c],[1,2,3],X).
```

```
X = [[a,1],[b,2],[c,3]]
```

```
?- combine2([foo,bar,yip,yup],[glub,glab,glib,glob],Result).
```

```
Result = [[foo,glub],[bar,glab],[yip,glib],[yup,glob]]
```

```
combine2([],[],[]).
```

```
combine2([X|Xs],[Y|Ys],[[X,Y]|Zs]) :- combine2(Xs,Ys,Zs).
```

3. Write a 3-place predicate `combine3` which takes three lists as arguments and combines the elements of the first two lists into the third as follows:

```
?- combine3([a,b,c],[1,2,3],X).
```

```
X = [join(a,1),join(b,2),join(c,3)]
```

```
?- combine3([foo,bar,yip,yup],[glub,glab,glib,glob],R).
```

```
R = [join(foo,glub),join(bar,glab),join(yip,glib),join(yup,glob)]
```

```
combine3([],[],[]).
```

```
combine3([X|Xs],[Y|Ys],[join(X,Y)|Zs]) :- combine3(Xs,Ys,Zs).
```

4. Write a predicate `mysubset/2` that takes two lists as arguments and checks whether the first list is a subset of the second.

```
mysubset([], _).
```

```
mysubset([X|Xs], Y) :- member(X, Y), mysubset(Xs, Y).
```

5. Write a predicate `mysuperset/2` that takes two lists as arguments and checks whether the first list is a superset of the second.

```
mysuperset(_, []).
```

```
mysuperset(X, [Y|Ys]) :- member(Y, X), mysuperset(X, Ys).
```