Exercise 1 (Navigation at Campus - 15 pts).

1.1) route_by_bike(F, T) :- bike(F, T, M) ; bike(F, X, M), route_by_bike(X, T).

1.2) route(F, T) :- bike(F, T, Z) ; walk(F, T, Z) ; (bike(F, X, Z) ; walk(F, X, Z)), route(X, T).

1.3) route(F, T, M) :- bike(F, T, M) ; walk(F, T, M) ; (bike(F, X, M1) ; walk(F, X, M1)), route(X, T, M2), M is M1 + M2.

Exercise 2 (Blocksworld - 15 pts).

2.1) X = b,
Y = a ;
X = c,
Y = b ;
X = c,
Y = a ;

2.2) atLeastThree(X) :- on(X,Y), on(Y,Z), on(Z,A).

2.3) exactlyThree(X) :- on(X,Y), on(Y,Z), on(Z,A), not(on(A,_)).

2.4) exactlyThreeTower(X) :- tower([X|T]), length(T,3).

Exercise 3 (A small program - 10 pts).

```
[trace] 3 ?- xyz([a,b],[a],[b]).
   Call: (10) xyz([a, b], [a], [b]) ? creep
   Call: (11) xy([a, b], [], [a], [b]) ? creep
   Call: (12) xy([b], [_15400], [], [b]) ? creep
   Exit: (12) xy([b], [b], [], [b]) ? creep
   Exit: (11) xy([a, b], [], [a], [b]) ? creep
   Exit: (10) xyz([a, b], [a], [b]) ? creep
true .
```

L splits into two seperate lists of equal length. If the length of L is odd, then the first list is one element longer.

Exercise 4 (Lists - 10 pts).

sum(A,B,C):- permutation(P, A), append(B,C,P), sum_list(B, X), sum_list(C,X).

Exercise 5 (Recursive predicate - 15 pts).

sum([], 0).

sum([Head|Tail], Sum) :- sum(Tail,TailSum), Sum is Head + TailSum.

mean(L,M) :- sum(L, Sum), length(L,Len), M is Sum//Len.

Exercise 6 (Movie database - 35 pts).

6.1) listMovies(L) :- findall(Title, movie(Title,_,_,_,_,_), L).

6.2) listMoviesByName(L1) :- listMovies(L2), sort(L2, L1).

6.3) listMoviesByGenre(G, L) :- findall(Title, (movie(Title,_,Genre,_,_,_), member(G,Genre)), L).

6.4) listMoviesByRank(L, S) :- findall(Rating-Title,movie(Title,_,_,_,_,Rating), R), sort(1, @>=, R, Sort), pairs_values(Sort, S).

6.5) numberMovies(L, G, Count) :- listMoviesByGenre(G,R), length(R, Count).