In Question 4 of Exercise 4 (last exercise on Prolog) you define a predicate `path(S, T, P)` that returns acyclic paths from S to T. In this homework you have to build upon your solution for Question 1. Consider the following weighted graph (all edges are bidirectional):

```
Figure 1: Base graph.
```

In particular, in this homework you need to take the knowledge base found in `basegraph.pl` which models the graph of Figure 1, and do the following:

1. write a predicate `path(S, T, P, L)` which returns all acyclic paths from S to T along with their length L.

2. write a predicate `spath(S, T, P, L)` which employs `path(S, T, P, L)` and returns the shortest path from S to T. For example, `spath(s, t, P, L)` must return:

```
?- spath(s, t, P, L).
P = [edge(s, n3, 3), edge(n3, n5, 3), edge(n5, t, 2)]
L = 8.
?- 
```

3. write a predicate `enumpaths(S, T, U, P, L)` which return true if path P is a path from S to T and are less than U percent longer than the shortest path from S to T. For example, `enumpaths(s, t, 0.5, P, L)` must return:
?- enumpaths(s,t,0.5,P,L).
P = [edge(s, n2, 4), edge(n2, n4, 5), edge(n4, t, 2)],
L = 11 ;
P = [edge(s, n3, 3), edge(n3, n1, 2), edge(n1, t, 6)],
L = 11 ;
P = [edge(s, n3, 3), edge(n3, n4, 5), edge(n4, t, 2)],
L = 10 ;
P = [edge(s, n3, 3), edge(n3, n4, 5), edge(n4, n5, 1), edge(n5, t, 2)],
L = 11 ;
P = [edge(s, n3, 3), edge(n3, n5, 3), edge(n5, t, 2)],
L = 8 ;
P = [edge(s, n3, 3), edge(n3, n5, 3), edge(n5, n4, 1), edge(n4, t, 2)],
L = 9 ;
false.
?-