```
station(a).
station(b).
station(c).
station(d).
station(e).
station(f).
station(g).
station(h).
station(i).
station(j).
station(k).
station(l).
station(m).
station(n).
station(o).
station(p).
station(q).

line(red).
line(green).
line(blue).
line(purple).
line(yellow).

stop(red,1,a).
stop(red,2,c).
stop(red,3,e).
stop(red,4,i).
stop(red,5,m).
stop(red,6,q).

stop(green,1,g).
stop(green,3,c).
stop(green,5,h).
stop(green,7,p).
stop(green,2,b).
stop(green,4,e).
stop(green,6,l).

stop(blue,6,k).
stop(blue,1,d).
stop(blue,5,j).
stop(blue,3,i).
stop(blue,2,h).
stop(blue,4,m).

stop(purple,5,n).
stop(purple,4,j).
stop(purple,3,i).
stop(purple,2,l).
stop(purple,1,o).

stop(yellow,5,g).
stop(yellow,3,i).
stop(yellow,7,n).
stop(yellow,1,o).
stop(yellow,6,k).
stop(yellow,4,f).
stop(yellow,2,l).
stop(yellow,8,q).

multiple_lines(S):- stop(X,_,S), stop(Z,_,S),\+Z=X.

notmaximum(C,N,L):- stop(C, N,L), stop(C,F,E), N<F.
notminimum(C,N,L):- stop(C, N,L), stop(C,F,E), N>F.
termini(C,S1,S2):-  stop(C,R,S1), \+notminimum(C,R,S1).
termini(C,S1,S2):-  stop(C,R,S2), \+notmaximum(C,R,S2).

highestvalue(C,VALUE):-termini(C,1,S2),stop(C,VALUE,S2).
orderednumlist(L,List):-highestvalue(L,Value), numlist(1,Value,List).
list_stops(C,List):- orderednumlist(C,Olist), maplist(stop(C),Olist,List).

```

```
73  edge(C,X,Y):- stop(C,M,X), stop(C,N,Y), N>M.
74  path(X,Y,Path):-pathBuilderHelper(X,Y,[],Path).
75  pathBuilderHelper(X,Y,VISITED,Path):-edge(C,X,Y),Path =
    [segment(C,X,Y)],segment_adds_cycle(segment(C,X,Y),VISITED).
76  pathBuilderHelper(X,Y,VISITED,Path):-edge(C,X,Z),segment_adds_cycle(segment(C,X,Z),VIS
    ITED),
    pathBuilderHelper(Z,Y,[segment(C,X,Z)|VISITED],ZYPath),Path=[segment(C,X,Z)|ZYPath],\+
    member(segment(C,_,_),ZYPath).
77
78  segment_edge(C,X,Y):- stop(C,M,X), stop(C,N,Y), N is M+1.
79  segment_to_path(segment(C,X,Y),Path):- segment_to_path_helper(C,X,Y,Path).
80  segment_to_path_helper(C,X,Y,Path):- segment_edge(C,X,Y),Path = [X].
81  segment_to_path_helper(C,X,Y,Path):- segment_edge(C,X,Z),
    segment_to_path_helper(C,Z,Y,ZYPath), Path=[X|ZYPath].
82
83  stations_traversed(Path,Set):- maplist(segment_to_path(),Path,Setlists),
    append(Setlists,Set).
84  stations_traversed(segment(C,S1,S2),Set):- segment_to_path(segment(C,S1,S2),Set).
85  compare_list(List1, List2):-intersection(List1,List2,X), \+dif(X,[]).
86  segment_adds_cycle(segment(C,S1,S2),Path):-stations_traversed(segment(C,S1,S2),Segment
    set), stations_traversed(Path,Listset), compare_list(Segmentset,Listset).
87
88  minimum_line_changes(S1,S2,Smallest):-findall(List,path(S1,S2,List),ListofLists),
    smallest_path(ListofLists,Smallest).
89  smallest_path(List,Size):- maplist(length(),List,Sizes),min_list(Sizes,Size).
90  minimum_path(S1,S2,Path):- minimum_line_changes(S1,S2,Linechanges),
    path(S1,S2,Path), length(Path,X), X==Linechanges.
91
```