Networks Routing Assignment Tests

NOTE: Project utilizes two-way edges not directional edges

- Part 1 - Setup and calculating cost of a packet [30 Marks]
  Get path function was tested using the provided example router setup and was used to search for every node which all returned the correct cost and path as shown below.

```
Start: a
End: a
Path: a
Cost: 0

Start: a
End: b
Path: a->b
Cost: 7

Start: a
End: c
Path: a->c
Cost: 9

Start: a
End: d
Path: a->c->d
Cost: 20

Start: a
End: e
Path: a->c->f->e
Cost: 20

Start: a
End: f
Path: a->c->f
Cost: 11
```

- Part 2 - Print routing table for router [30 Marks]
  Print routing table function was implemented using the pandas dataframe as referenced in the brief. Function was called for router “a” below.

```
  from to  cost        path
0    a  a     0           a
1    a  b     7        a->b
2    a  c     9        a->c
3    a  f    11     a->c->f
4    a  d    20     a->c->d
5    a  e    20  a->c->f->e
```

- Part 3 - Remove router [10 Marks]

  Remove router function was implemented removes the router and any related edges from our graph. Seen below is the output provided by removing the c router from our network.

```
  from to  cost     path
0    a  a     0        a
1    a  b     7     a->b
2    a  f    14     a->f
3    a  d    22  a->b->d
4    a  e    23  a->f->e
```

- Part 4 - Multiple Routers [20 Marks]

  Multiple routers using a shared graph was implemented, each with their own calculated routing table. Removing a router successfully clears it from all routing tables.

```
router = Router('a', graph)
router_two = Router('b', graph)
router.print_routing_table()
  from to  cost        path
0    a  a     0           a
1    a  b     7        a->b
2    a  c     9        a->c
3    a  f    11     a->c->f
4    a  d    20     a->c->d
5    a  e    20  a->c->f->e
router_two.print_routing_table()
  from to  cost        path
0    b  a     7        b->a
1    b  b     0           b
2    b  c    10        b->c
3    b  f    12     b->c->f
4    b  d    15        b->d
5    b  e    21  b->c->f->e
router.remove_router('c')
  from to  cost     path
0    a  a     0        a
1    a  b     7     a->b
2    a  f    14     a->f
3    a  d    22  a->b->d
4    a  e    23  a->f->e
router_two.print_routing_table()
  from to  cost     path
0    b  a     7     b->a
1    b  b     0        b
2    b  f    21  b->a->f
3    b  d    15     b->d
4    b  e    21  b->d->e
```

- Nice little extras [10 Marks]

  I chose to implement NetworkX graph visualizations into my assignment. By calling graph.print_graph() you get an image of your router network as shown below.