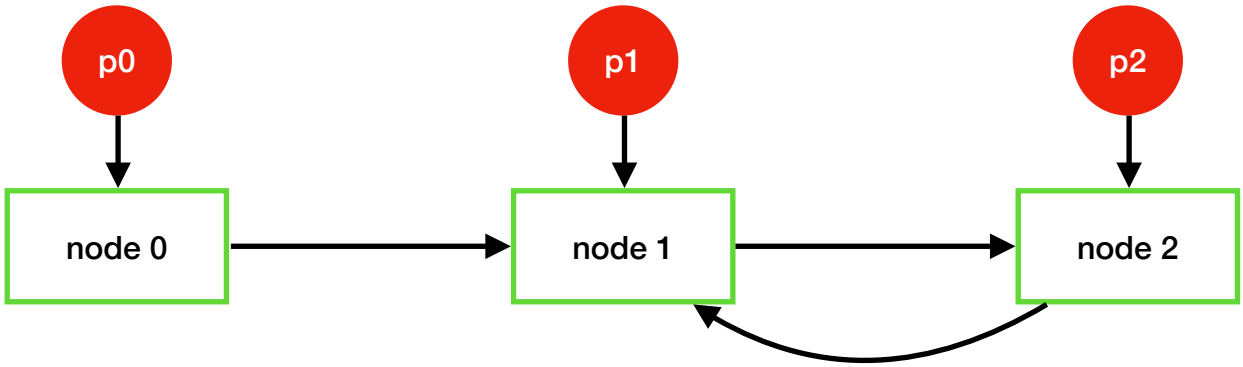
Week 8

Exercise 2

```
Node *p0 = new Node('0');
Node *p1 = p0->next = new Node('1');
Node *p2 = p1->next = new Node('2', p1);
```



Exercise 4

Esitetty push-metodin toteutus liitetyn pinon osalta käyttää automaattimuistin varauksen sijaan dynaamista muistin varausta (heap-muistia). Ongelma tässä lähestymistavassa on se, että new_top -niminen Node-objekti luodaan funktion sisällä ja on siten paikallinen funktion ulkopuolella. Kun funktio päättyy, new_top-objekti tuhoutuu, ja top_node jää osoittamaan vapautetun muistialueen osoitetta.
Oikea tapa toteuttaa push-metodi liitetyn pinon kohdalla on käyttää dynaamista muistinvarausta (new-avainsanaa). Alla on esimerkki oikeasta toteutuksesta:

Oikean lainen toteutus olisi;

```
Error_code Stack::push(const Stack_entry &item)
{
    Node *new_top = new Node(item, top_node);
    if (new_top == nullptr) {
        return overflow; // Tarkista, epäonnistuiko muistin varaus
    }

    top_node = new_top;
    return success;
}
```

Tässä tarkistetaan, epäonnistuiko muistinvaraus (jos new_top on nullptr). Jos muistinvaraaminen epäonnistuu, palautetaan overflow, mikä osoittaa, että pinon koko ei riitä uuden alkion lisäämiseen.