

1. Ohjelmoinnin perusteiden kertausta. Selvitä alla olevan ohjelman toiminta.

```
/**
 * This file is part of ATK3/excercises
 * Created by Heikki Laitala 23.1 2008
 *
 */

#include <iostream>

#define MAX 100 //Yleensä ottaen defineitä pitäisi
               //pyrkä välttämään, jos niitä haluaa käyttää
               // niin on suositeltavaa määrittää yksi
               // otsikko-tiedosto jossa esittelet kaikki
               // ohjelmassa käytettävät define-arvot.

/**
 * Tässä tehtävässä sinun tulee selvittää,
 * oko välitulostusta tai ohjelmointiympäristön
 * debuggeria käyttäen apina- sekä gorilla-
 * muuttujan arvo eri kohdissa ohjelman suoritusta.
 * Yritä ymmärtää, mitä ohjelmassa tapahtuu.
 */

int main()
{
    /**
     * Peukalosääntö (rule-of-thumb) muuttujien
     * esittelyssä: Esittele (luo) muuttuja siellä,
     * missä niitä tarvitaan eikä esim. ohjelman alussa.
     * Jos on mahdollista niin vapauta dynaamiset
     * (kekkoon luodut) muuttujat heti sen jälkeen,
     * kun niitä ei tarvitse.
     */

    double apina = 0.1;
```

```

int kirahvi = 12;
int gorilla = 5;

if(kirahvi != 12 || gorilla == 5 )
{
    apina = 1.0;

double kibboni = 2.1;

if( MAX == 200 && kirahvi == 12)
{
    apina = apina - kibboni;
}

else
{
    apina = apina + kibboni;
}

kirahvi = 11;

bool apina_on_elain = true;

if(kibboni < 2.2 && apina_on_elain)
{
    //Mitä tässä tapahtuu?
    ++gorilla;
}
//Mikä on gorilla muuttujan arvo tässä kohden?

//Tiukkoja kysymyksiä lukijalle: Miksi on
//järkevämpää
//käyttää (muiden kuin perustietotyyppien) kanssa
//ennenmin preincrement- kuin postincrement-
//operaattoria?
//preincrement : ++gorilla
//postincrement: gorilla++;

}
else
{

```

```
apina = 12.5;
kirahvi = 2;

int oranki = 2;
kirahvi = oranki +1;
}
```

```
//Mikä on apina muuttujan arvo tässä kohden ohjelman
//suoritusta?
```

```
switch(gorilla)
{
    case 1:
    {
        gorilla = kirahvi;
        break;
    }
    case 2:
    {
        gorilla -= kirahvi;
        --apina;
        break;
    }
    default:
    {
        gorilla += 2;
        break;
    }
}
```

```
//Mikä on gorilla- sekä apinan-muuttujan arvo tässä
//kohden ohjelman suoritusta?
```

```
double oranki = 2.0;

for(int k = 0; k < 10; ++k)
{
    ++gorilla;
}
```

```

if( k % 2)
{
    --gorilla;
}
else if( k == 3)
{
    apina = apina + 0.1;
}
else
{
    oranki = apina+1.0;
}
}

apina = oranki;

// Mikä on gorilla- sekä apina-muuttujan arvo tässä
// kohden ohjelman suoritusta?

//Huomaa tyyppimuunnos!
kirahvi = int(apina);

//Kun tyyppimuunnetaan muita kuin perustietotyypppejä
//c++-kielessä niin pitää käyttää static_cast-
//operaattoria:
//Esim:
//int kirahvi = static_cast<int>(apina);

//Mikä on kirahvin arvo tässä kohden ohjelman
//suoritusta?

return 0;
}

```

2. Kirjoita LC -piiriä kuvaava luokka. Siinä tulisi olla ainakin metodit:

- `setL( float )` induktanssin L asettamiseksi
- `getL()` induktanssin kysymiseksi
- `setC( float )` kapasitanssin C asettamiseksi
- `getC()` kapasitanssin kysymiseksi
- `resFreq()` resonanssitaajuuden $1/\sqrt{LC}$ laskemiseksi
- `show()` kapasitanssin ja induktanssin näyttämiseksi.

Kirjoita myös pääohjelma, jolla voit testata luokan toimintaa.

3. Suunnittele ja toteuta toisen kertaluvun lineaarisen vakiokertoimisen differentiaaliyhtälön

$$y''(t) + \omega^2 y(t) = 0; \quad y(0) = y_0 \text{ ja } y'(0) = v_0$$

käsittelyyn sopiva luokka. Muistin virkistämiseksi kerrottakoon, että yhtälön yleinen ratkaisu on

$$y(t) = A \sin \omega t + B \cos \omega t,$$

jossa kertoimet A ja B määräytyvät alkuarvoista. Luokan olioiden tulisi vastata ainakin kysymyksiin paikan $y(t)$ ja nopeuden $y'(t)$ arvoista ajan hetkellä t .