Fyrirlestraræfing 7

1. Hvað myndi gerast í lykkjunni sem á að prenta út 10 "Halló" ef við settum "**i = i - 1**" í stað "**i = i + 1**"?

   *Þá myndi gildi breytunnar **i** lækka sífellt, þar til við fengjum yfirflæði (eða kannski frekar undirflæði!) þegar stærsta möguleg mínustala (-2,147,483,648) í **int** breyttist í stærstu mögulegu plústölu (2,147,483,647) í **int**. En það væri ekki fyrr en eftir um 2 þús. milljónir ítranir!*

2. Hvert er vandamálið við lykkjuna á myndinni:

```
int i = 1;
while (i <= 10)
   System.out.println(i + "th Hello");
   i = i + 1;
```

   *Það vantar slaufusviga, þannig að það er aðeins **println**-skipunin sem er í lykkjunni. Gildi breytunnar **i** breytist aldrei, svo þetta er endalaus lykkja.*

3. Skrifið lykkju sem prentar út *sin*-fallið á bilinu 0.0 til 1.0 með upphækkun 0.1:

```
double x = 0.0;
while (x <= 1.0) {
    System.out.println(x + ": " + Math.sin(x));
    x += 0.1;
}
```

eða

```
for (double x = 0.0; x <= 1.0; x += 0.1) {
    System.out.println(x + ": " + Math.sin(x));
}
```

4. Skrifið **for**-lykkju sem reiknar 1*2*3*...*N, þ.e. N!

```
double prod = 1.0;
for (int i = 2; i <= N; i++) {
    prod *= i;
}
```

5. Ef upphafsgildi **a** er 5 og **b** er 2, hvert er gildi þeirra í lok eftirfarandi kóða?

```
a = b++ - 1;
b -= ++a;
```

   *Í fyrri línunni fær breytan **a** gildið 2-1, eða 1. Í þeirri línu er **b** líka hækkað uppí 3. Í seinni línunni er **a** hækkað um 1, uppí 2 og **b** er lækkað um það gildi, eða 3-2 = 1. Lokagildi **a** er því 2, en lokagildi **b** er 1.*

6. Hvert er gildi **j** eftir að eftirfarandi skipun hefur verið framkvæmd?

```
for (i = 0, j = 0; i < 10; i++)
    j += i;
```

*Breytan **j** endar með gildið 45, því gildi **i** er lagt við hana í hverri ítrun og **i** fær gildin 0, 1, 2, .., 9. Summa þeirra er 45.*

Fyrirlestraræfing 8

1. Sýnið **for**-lykkju til að reikna fyrstu *N* liðina í röðinni $\frac{1}{4}+\frac{1}{16}+\frac{1}{64}+...+\frac{1}{4^N}$

```
double sum = 0.0;
for (int i = 1; i <= N; i++) {
    sum += Math.pow(0.25, i);
}
```

2. Hvernig er tugatalan 37 skrifuð sem tvíundartala?

Finnum fyrst hæsta heila veldi á 2 sem er minna en 37. Það er 32 = 2^5. Höfum þá 1?????. Sjáum síðan að 32+16 er of stórt, sömuleiðis 32+8, en 32+4(=2^2) = 36 er í lagi, svo nú höfum við 1001??. Sjáum að 36+2 er of stórt, en 36+1(=2^0) = 37 passar. Svo tvíundartalan er 100101.

3. Sýnið **for**-lykkju sem skrifar út "girðingu" með *N* staurum eins og sést hér fyrir neðan

|=|=|=|=|=|=|

```
System.out.print("|");
for (int i = 1; i < N; i++) {
    System.out.print("=|");
}
System.out.println();
```

4. Í síðustu **println**-skipun forritsins **Gambler** er segðin **1.0*bets/T**. Hvers vegna er margfaldað með 1.0?

*Það er vegna þess að breyturnar **bets** og **T** eru báðar **int**-breytur og við fengjum því heiltöludeilingu ef bara stæði **bets/T**. Önnur leið er að nota kast og skrifa þá t.d. **(double)bets/T**. Þá er gildi **bets** breytunnar breytt í kommutölu og framkvæmd er kommutöludeiling.*

5. í **Gambler** leiknum, ef þið byrjið með $10 og viljið hætta með $100 hvað megið þið búast við að þurfa að veðja oft?

   Samkvæmt stærðfræðiformúlunni eru þetta 10(100-10) = 10*90 = 900 skipti.*

6. Skrifið **do-while** setningu sem notar **Math.random()** til að finna slembitölu á bilinu [0, 0.5]?

```
double r;
do {
    r = Math.random();
} while ( r > 0.5 );
```

*Lykkjan heldur áfram að biðja um nýja slembitölu, á meðan talan er > 0.5. Athugið að nú er bil slembitalnanna sem við fáum lokaða bilið frá 0.0 til 0.5 (þ.e. báðar meðtaldar). Ef við hefðum sett skilyrðið sem **r >= 0.5**, þá hefðum við fengið bilið [0, 0.5), en það hefði verið mun auðveldara að fá það með því að margfalda bara **r** með 0.5.*