アーキテクチャ　第３回　　資料１

1） 固定小数点の原理

各桁を展開した形に書いてみよ　1.101_2 ＝ ……　　101.11_2 ＝ ……

2） 2のべき乗の値を覚えよ

2の(－1)乗＝……　(－2)乗＝……　(－3)乗＝……　(－4)乗＝……　(－5)乗＝……

3） 次の(2進⇔10進)変換をせよ

（ア）　1.0101_2 ＝ ……$_{10}$　　（イ）　11.1001_2 ＝ ……$_{10}$

（ウ）　0.25_{10} ＝ ……$_2$　　（エ）　0.5625_{10} ＝ ……$_2$

（オ）　0.1_{10} ＝ ……$_2$

5） 有効数字と、 科学的表記法(指数表記)について

（ア） 60.8 の有効数字は……桁である　　（イ） 0.000608 の有効数字は……桁である

（ウ） 60800000 の有効数字は……桁である

（エ） (ア)～(ウ)を科学的表記法(指数表記)によって表すと

60.8 ＝……　0.000608 ＝……　60800000 ＝……

6） 浮動小数点表現は、科学的表記法を2進に適用したものと考えられる。 仮数部＝1.101、指数部＝11 の表す数は、2進の科学的表記で ……と書け、実際の値は10進で……になる。

（注） コンピュータ上での浮動小数点数の表記は、仮数部も指数部も、単純に2進表記したものではなく、符号を表す工夫や、ビット数を減らす工夫がされている。

7） 単精度浮動小数(32ビット)と、倍精度浮動小数(64ビット)の表すことのできる、10進で見た時の有効数字のおよその桁数はいくつか？ 単精度の場合……桁 倍精度の場合……桁。 但し、

単精度浮動小数は、符号1ビット、指数部符号付8ビット(－126～127)、仮数部23ビット であり、

倍精度浮動小数は、符号1ビット、指数部符号付11ビット(－1022～1023)、仮数部52ビット とする。

8）（討論問題） 浮動小数点表現は、固定小数点表現と比較して、どういう利点があるか？　どういう欠点があるか？

……

……

……

……

……

アーキテクチャ　第３回　　資料２

コンピュータ上では、数値は、有限の桁数の数字として表されるため、はみ出す部分については誤差(切り捨てにせよ切り上げにせよ)となる。

9)

(ア) 16ビット符号付き整数の最大値、最小値は何か？　　(その範囲を超える数は、16ビット整数の中に収まらない)

(イ) 16ビット符号なし整数の最大値、最小値は何か？

(ウ) (討論問題)
右のプログラムが int 型(符号付の32ビット整数型)のオーバーフローを起こしていることを説明せよ

プログラム
```
class Myfact {
  public static void main(String[] args) {
    int i, fact;
      fact = 1;
      for (i=2; i<=25; i++) {
        fact = fact * i;
        System.out.println("i: " +  i +
           "  fact: " + fact);
      }
   }
}
```

実行結果
```
i: 2  fact: 2
i: 3  fact: 6
i: 4  fact: 24
i: 5  fact: 120
i: 6  fact: 720
i: 7  fact: 5040
i: 8  fact: 40320
i: 9  fact: 362880
i: 10  fact: 3628800
i: 11  fact: 39916800
i: 12  fact: 479001600
i: 13  fact: 1932053504
i: 14  fact: 1278945280
i: 15  fact: 2004310016
i: 16  fact: 2004189184
i: 17  fact: -288522240
i: 18  fact: -898433024
i: 19  fact: 109641728
i: 20  fact: -2102132736
i: 21  fact: -1195114496
i: 22  fact: -522715136
i: 23  fact: 862453760
i: 24  fact: -775946240
i: 25  fact: 2076180480
```

10) (討論問題) 次の問題を解くプログラムを作りたい。

ア) 桁落ちが起こる可能性があることを説明せよ

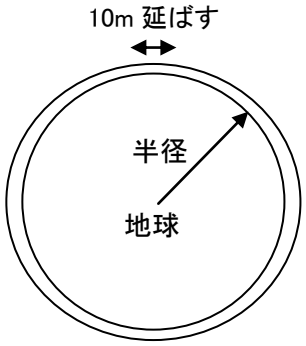
イ) 実際、単精度浮動小数(float 型、仮数部が 23 ビット＝10 進で約8桁)で実行すると桁落ちが起こる。 下記のプログラム例を参照し、確認せよ。

ウ) 解決策を考えてみよ。 もちろん、倍精度浮動小数(double 型、仮数部が 52 ビット＝10 進で約15桁半)に変えるというのはアリだが、そうでない方法を考えてみよ。

地球の赤道上にリボンを1周させる。 そのリボンを 10 m 継ぎ足して巻くと、リボンは少しだけ緩むはずである。
地球の半径を 6 378 136.6 m とすると、このリボンは地上どれだけの高さに巻かれることになるか？ 但し地球は完全な球体とし、リボンは同じ高さで1周させるものとする。

```
class Myradius {
    public static void main(String[] args) {
        float radius, circumference, ribbon, ribbonradius, pi, height;
        pi = (float) 3.14159265358979;
        radius = (float) 6378136.6;
        circumference = (float) 2.0 * pi * radius;
        ribbon = circumference + (float) 10.0;
        ribbonradius = ribbon / (float) 2.0 / pi;
        height = ribbonradius - radius;
        System.out.printf(
           " radius: %f¥n ribbonfadius: %f¥n height:  %f¥n",
              radius, ribbonradius, height);
    }
}
```

このプログラムでは、定数をわざわざ float に型変換(キャスト)している。



実行結果
```
radius: 6378136.500000
ribbonfadius: 6378138.000000
height: 1.500000
```