Supervision work
Supervision 17

Tudor Avram
Homerton College – tma33@cam.ac.uk

May 3, 2016

1 Exercise 1

BCD (binary-coded decimal) is a class of binary encoding of decimal numbers used in computing and electronic systems. Every decimal digit is represented by a fixed number of bits (usually four or eight). Special bit patterns are sometimes used for sign or other indicators, such as overflow, etc.
Pocket calculators use BCD as they usually only have digital logic and no microprocessor. This way, the manipulation of the numerical data can be largely simplified by assigning a separate circuit to every digit.

2 Exercise 3

In theory, the base-16 representation would give us a wider range than the classic (binary) one. In the single-precision case (32-bit), though, the precision can be an issue. For the base-16 representation, there can be only 21 bits of precision when converted to binary, while for the classic one there are 24. Therefore, calculations can go worse on base-16 than they do in binary (which is already pretty bad). Thus, I think that using base-16 representation is not a practical idea.

3 Exercise 10

(a) 2.2 –>2
3.5 –>4
4.5 –>4
5.6 –>6
10.9 –>11

(b) 111.11 –>111
111.101 –>1000

101.10 –>101
110.1 –>110

(c) 1.2345e5 –>1.23e5
2.255e-10 –>2.26e-10

4 Exercise 11

In my opinion, even though integers are a subset of the doubles, the fact that we use both is far from silly. This is because most of the time we only have to use integers and we need to be exact in our calculations, so we can't take the risk of using floating points(not even double-precision one). A good example in this case are iterations, that are really easy to do with integers, but are very risky when using a floating point as an iterator.

5 Exercise 13

(a) 00 00 00 00 = 0000 0000 0000 0000 0000 0000 0000 0000 = +0
(b) 80 00 00 00 = 1000 0000 0000 0000 0000 0000 0000 0000 = −0
(c) BF 01 00 00 = 1011 1111 0000 0001 0000 0000 0000 0000 = −0.5
(d) 3F C0 00 00 = 0011 1111 1100 0000 0000 0000 0000 0000 = 1.5
(e) 04 04 04 00 = 0000 0100 0000 0100 0000 0100 0000 0000 = 263168×2^{-119}

6 Exercise 14

$\log_{10}(2) = 0.3$ Therefore, the number of digits in the decimal representation is given by $[\log_{10}(2)] + 1$, while the number of bits in the binary representation is given by $\log_2(n)$, for whatever n.

7 Exercise 15

```
public static void main(String[] args) {
        float a = (float) 1.0;
        float b = (float) 1e-8;
        float c = (float) 1.0;
        System.out.println(1.0*1e-8+1.0); // Gives 1.00000001
        float t = a*b;
        System.out.println(t+c); // Gives 1.0
}
```

8 Exercise 16

Machine epsilon is the difference between 1.0 and the smallest representable number which is greater than 1. A strategy to determine the machine epsilon would be to "simulate" a binary search. (i.e. it's more an approximating than determining it) :

```
double value = 1.5;
double epsilon = value - 1.0
while (value - 1.0 != 0.0) {
        epsilon = value - 1.0;
        value = (1.0 + value)*0.5;
}
```