Homework 2

20125071 – Bùi Lê Gia Cát

1. P2.84

```
int float_le(float x, float y){
    unsigned ux = f2u(x);
    unsigned uy = f2u(y);
    /* Get the sign bits */
    unsigned sx = ux >> 31;
    unsigned sy = uy >> 31;
    /* Give an expression using only ux, uy, sx, and sy */
    return ((!(ux<<1) && !(uy<<1)) ||
            (sx && !sy) || //x is neg and y is pos
            (sx && sy && ux>=uy) ||
            (!sx && !sy && ux<=uy));
}
```

2. P2.88

FORMAT A		FORMAT B	
Bits	Value	Bits	Value
1 01111 001	–9/8	1 0111 0010	–9/8
0 10110 011	176	0 1110 0110	176
1 00111 010	–5/2048	1 0000 0000	–1/128
0 00000 111	7/32763	0 0000 0001	1/1024
1 11100 000	–8192	1 1110 1111	–248
0 10111 100	384	0 1111 0000	+∞

3. P2.91

A. 11.00100100000111111011011

B. 11.001001001(001)

C. These two approximations to π diverge at the 9th to the right of the binary point.

4. P2.87

Description	Hex	M	E	V	D
– 0	8000	0	– 15	– 0	– 0.0
Smallest value > 2	0401	1025/1024	1	1025.2^{-9}	2.001953
512	6000	0	9	512	512.0
Largest denormalized	03FF	1023/1024	-14	1023.2^{-24}	0.000061
– ∞	FC00	---	---	– ∞	– ∞
Number with hex representation 3BB0	3BB0	59/64	-1	123.2^{-7}	0.960938

5. P2.92

```
float_bits float_negate(float_bits f){
    unsigned s=f>>31;
    unsigned e=(f>>23)&0xFF;
    unsigned frac=f&0x7FFFFF;

    if (e==0xFF && frac!=0){
        return f;
    }

    return ((~s)<<31) | (e<<23) | (frac);
}
```

6. P2.94

```
float_bits float_twice(float_bits f){
    float_bits s=f>>31;
    float_bits e=(f>>23) & 0xFF;
    float_bits frac=f & 0x7FFFFF;

    if (e&0xFF){
        return f;
    }
    else if (e==0xFF-1){
        frac=0;
    }
    return (s<<31) | (e+1)<<23 | frac;
}
```

7. P2.95

```
float_bits float_half(float_bits f){
    unsigned s=f>>31;
    unsigned e=(f>>23)&0xFF;
    unsigned frac=f&0x7FFFFF;

    if (e==0xFF){
        return f;
    }
    else if (e==0x0){
        return 0;
    }

    return (s<<31) | (e-1)<<23 | frac;
}
```

8. P2.96

```c
int float_f2i(float_bits f){
    unsigned s=f>>31;
    unsigned e=(f>>23)&0xFF;
    unsigned m=f&0x7FFFFF;

    unsigned bias=0x7F;

    if (e<bias){
        return 0;
    }

    int result;
    if (e>=31+bias || e==0xFF){
        result= 0x80000000;
    }
    else{
        e-=bias;
        m |=0x80000000;

        if (e>23){
            result = m<<(e-23);
        }
        else {
            result = m>>(23-e);
        }
    }
    if (s) return -result;
    else return result;
}
```

9. P2.97

```c
int bitLength(int i){
    i>>=1;
    int l=0;
    while (i){
        ++l;
        i>>=1;
    }
    return l;
}

float_bits float_i2f(int i){
        unsigned s=i>>31;
    int length=bitLength(i);
    unsigned e=length+127;

    i&=(1<<length)-1;

    float_bits result = s<<31;
    result |=(e<<23);
    result |= (i<<(23-length));

    return result;
}
```