**实验2 make & 位级运算**

**徐薪-519021910726**

**一、 make**

1. gcc

2. make

3. make clean

4. 11

5. 14

6. FOO = the value of the variable “FOO”. And using $(FOO) to replace the value of “FOO”.

**二、 位级运算、数的编码**

1.

int allOddBits(int x) {

//your codes Here

int odd=0xAAAAAAAA;

int y;

y=x&odd;

return !(y^odd);

}

2.

int isLessOrEqual(int x, int y) {

//your codes here

if(((x&~y)>>31)&1) return 1;

if(!(x^y)) return 1;

if(((~x&y)>>31)&1) return 0;

return !((y+~x+1)>>31);

}

3.

int logicalNeg(int x) {

//your codes here

return ~(x|(~x+1))>>31&1;

}

4.

unsigned floatScale2(unsigned uf) {

//your codes here

int e=(uf&0x7F800000)>>23;

int s=uf&0x80000000;

if(e==0) return (uf<<1)|s;

if(e==255) return uf;

++e;

if(e==255) return 0x7F800000|s;

return (uf&0x807FFFFF)|(e<<23);

}

5.

int floatFloat2Int(unsigned uf) {

//your codes here

int te=((uf&0x7F800000)>>23)-127;

int m=(uf|0xFF800000)^0xFF000000;

int s=uf>>31;

if(te<0) return 0;

if(te>31) return 0x80000000;

if(te>23) m=m<<(te-23);//32 all occupied

else m=m>>(23-te);

if(!((m>>31)^s)) return m;

else if(!s) return 0x80000000;

else return ~m+1;

}

And the output of test of bits.c is:

文本

描述已自动生成