

# 实践2 实验报告

陈文迪 519021910071

## 任务1 make

问题回答:

1. 本程序使用 gcc 编译器。
2. 可以使用 `make all` 命令编译全部程序。
3. 可以使用 `make clean` 命令删除全部编译结果。
4. 文件中第 10-11 行生成btest的目标文件。

```
btest: btest.c bits.c decl.c tests.c btest.h bits.h
$(CC) $(CFLAGS) $(LIBS) -o btest bits.c btest.c decl.c tests.c
```

5. 文件中第 13-14 行生成fshow的目标文件。

```
fshow: fshow.c
$(CC) $(CFLAGS) -o fshow fshow.c
```

6. 可以通过 `$(FOO)` 来引用变量。

## 任务2 位级运算、数的编码

1. **思路:** 先构造出0xAAAAAAAA, 再利用“与”运算得到结果。需要注意的是, 为了将结果转化为0和1, 我们可以利用~和!来实现。

```
int allOddBits(int x) {
    //your codes Here
    int a = (0xAA<<8)+0xAA;
    int b = (a<<16)+a;
    return (!(~x&b));
}
```

2. **思路:** 分类讨论。若两数符号位不同, 则可以直接比较符号位得到答案; 若两数符号位相同, 则可以用两数做差得到x-y, 这种情况不可能发生溢出, 因此可以通过判断差的符号来得到x与y的大小关系; 若所得差为0, 单独判断即可。

```
int isLessOrEqual(int x, int y) {
    //your codes here
    int sign_x = (x>>31)&1;
    int sign_y = (y>>31)&1;
    int if_diff_sign = (sign_x^sign_y);
    int diff = x + ~y + 1;
    int sign_diff = (diff>>31)&1;
    int if_zero = !(diff^0);
    return (if_diff_sign&sign_x) | ((!if_diff_sign)&(sign_diff | if_zero));
}
```

3. **思路：** 当一个整型不为0时，其补码与自身取“或”运算可以得到形如111...000...这样前缀为1的数，而0的补码仍然为000...，其最终结果仍然是000...，由此通过符号位即可实现逻辑非。

```
int logicalNeg(int x) {
    //your codes here
    int head = (((~x+1)|x)>>31)&1;
    return head^1;
}
```

4. **思路：** 本题主要难点在于细节较多，需要理解规范数和非规范数的表达规则，然后分类讨论修改阶数和尾数的值即可。

```
unsigned floatScale2(unsigned uf) {
    //your codes here
    int e_m = (uf)&((1<<31)-1);
    int e = (uf>>23)&((1<<8)-1);
    int m = (uf)&((1<<23)-1);
    int rest = uf - m;
    if(e==0xFF) return uf;
    if(!e_m) return uf;
    if(!e) return rest + (m<<1);
    return uf + (1<<23);
}
```

5. **思路：** 类似于第四题的处理，分类讨论。本题需要注意极小数的舍入和极大数的处理，细节较多。

```
int floatFloat2Int(unsigned uf) {
    //your codes here
    int inf = (1<<31);
    int sign = (uf>>31)&1;
    int e_m = (uf)&(inf-1);
    int e = (uf>>23)&((1<<8)-1);
    int m = (uf)&((1<<23)-1);
    int exp = e-127-23;
    int ans = (m+(1<<23));
    int sign_exp = (exp>>31)&1;
    if(sign_exp){
        exp = -exp;
        ans = ans>>exp;
        if((exp>=24)) return 0;
    }
    else{
        ans = ans<<exp;
        if((exp>=8)) return inf;
    }
    if(e==0xFF) return inf;
    if(!e_m) return 0;
    if(!e) return 0;
    if(!sign) return ans;
    else return -ans;
}
```

运行结果如下：

```
onlooker@MSI:~/Architecture/lab2/datalab$ ./dlc ./bits.c
onlooker@MSI:~/Architecture/lab2/datalab$ ./btest
```

Score	Rating	Errors	Function
2	2	0	allOddBits
4	4	0	isLessOrEqual
4	4	0	logicalNeg
5	5	0	floatScale2
5	5	0	floatFloat2Int

Total points: 20/20