ICS第二次作业

201300066 麻超

13

第一步结束后：$x \to a, y \to a\backslash\hat{}\ b$. *这里表示不出来异或的符号*

第二步结束后：$x \to b, y \to a\backslash\hat{}\ b$.

第三步结束后：$x \to b, y \to a$.

14

当len为偶数时，不会发生问题，如left,right为1和2时，执行交换，随后left,right分别变为2和1，退出循环。

但是当len为奇数时，最后一步会变为自己与自己交换，如此，在调用上一题函数时会发现每次操作自己与自己执行异或操作得到的总是0，改变了原来的数值。

可以将退出循环的条件改为`left<right`,如此当len为奇数时不会对最中间的数执行操作。

15

x	y	x^y	x&y	x|y	~x|~y	x&!y	x&&y	x||y	!x||!y	x&&~y
0x5F	0xA0	0xFF	0x00	0xFF	0xFF	0x00	0x01	0x01	0x00	0x01
0xC7	0xF0	0x37	0xC0	0xF7	0x3F	0x00	0x01	0x01	0x00	0x01
0x80	0x7F	0xFF	0x00	0xFF	0xFF	0x00	0x01	0x01	0x00	0x01
0x07	0x55	0x52	0x05	0x57	0xF5	0x00	0x01	0x01	0x00	0x01

16

1.x=(x>>(n-8))<<(n-8)

2.x=(x&0xFF)

3.x=x^(~(x&~0xFF))

4.x|0xFF

17

原值	机器数	二进制表示	真值
b8 01 00 00	000001B8H	+110111000B	440
14	14H	+10100B	20
58 fe ff ff	FFFFFE58H	-110101000B	-424

原值	机器数	二进制表示	真值
74 fe ff ff	FFFFFE74H	-110001100B	-396
44	44H	+1000100B	68
c8 fe ff ff	FFFFFEC8H	-110001100B	-312
10	10H	+10000B	16
0c	0CH	+1100B	12
ec fe ff ff	FFFFFEECH	-100010100B	-276
10	20H	+100000B	32

18

在本程序中，`size_t`的类型是`unsigned int`,而两个无符号整数相减后得到的值必然也被解释为无符号整数，所以其永远不能正确地判断，正确的办法是将第三行改为`return strlen(str1)>strlen(str2)`

21

`x<<=4`等价于`x=16x`，然后有`x_0=t,x=x-t`，所以`x=15x`，即M=15.

`y>>=2`等价于`y=y/4`,即N=4.

30

若高32位的乘积中每一位都与低32位的最高位相同，即不溢出，否则溢出

```
int ch_mul_overflow(int x,int y){
    long long sf=(long long)x*y;
    if (sf!=(int)sf){
        return 1;
    }
    else return 0;
}
```

31

不能。

这种方式虽然让`arraysize`的表示范围扩大，避免了其溢出，但调用`malloc`函数时，由于`uint`只有32位的表示范围，所以如果`arraysize>2^32-1`，那么同样会发生溢出。

可以在调用`malloc`函数之前先判断申请的空间大小是否大于`uint`的最大表示范围，如果是就返回-1表示不成功，否则再申请空间继续进行数组复制。

34

1. 非永真。若x=65534，则$x*x=2^{32}-2^{18}+4(\mod 2^{32})=-(2^{18}-4)$.
2. 非永真。当$x=-2^{31}$时，$x<0,x-1=7\text{FFFFFFFH},x-1>0,(x-1<0)||x>0$.不成立。
3. 永真。若$x>0$,则$-x$的符号位一定为1，即小于0，若$x=0$,则$-x=0$。所以前者为假时后者一定为真。
4. 非永真。$x=-2^{31},x/-x=80000000\text{H},-x<0$.此时两者都不成立。
5. 非永真。$x=0,x\&0xf!=15==0,(x<<28)<0==0$,两者同时为假。
6. 非永真。$x=-2^{31},y\neq-2^{31}$,此时结果为假。
7. 永假。
   $[-x]_{补}=\sim[x]_{补}+1,\sim[x]_{补}+\sim[y]_{补}=[-x]_{补}+[-y]_{补}-2,\sim[x+y]_{补}=[-x]_{补}+[-y]_{补}-1$
   .
8. 永真。$(\text{int})(\text{ux-uy})=[x-y]_{补}=[x]_{补}+[-y]_{补}=[-(y-x)]_{补}$.
9. 永真。右移总是向着无穷大方向取整。
10. 永真。乘2^k的结果和左移k位相同，无论结果是否溢出。
11. 非永真。$x=-1\vee y=-1,x>>2\vee y>>3=-1,x/4\vee y/8=0$.
12. 永真。$x*y,ux*uy$的低32位有完全一样的位序列。
13. 永真。无论带符号整数或者无符号整数，其在同一个加减法运算部件内运算，所以各自的机器数相同，两结果位序列相同。
14. 永真。$-y=\sim y+1,ux*uy=x*y$,所以左边为$x*(-y-1)+x*y=-x$.

35

1. 永真。IEEE 754标准符号和数值部分分开计算，无论结果是否溢出都不会影响乘积符号。
2. 非永真。当x的有效位数比float可表示的最大有效位数24更多时，会造成有效位数丢失，但double不会。
3. 非永真。$x+y$可能溢出，但$dx+dy$不会溢出。
4. 永真。double可以精确表示int型数据，所以尾数不会舍入，没有偏差。
5. 非永真。相乘的结果可能会发生舍入，造成偏差。
6. 非永真。dx,dy中只要有一个为0，另一个不为0就不相等。