

南京航空航天大学《计算机组成原理II课程设计》报告

- 姓名：曹伟思
- 班级：1617302
- 学号：161730213
- 报告阶段：Lab1
- 完成日期：2019.4.2
- 本次实验，我完成了所有内容。

目录

南京航空航天大学《计算机组成原理II课程设计》报告

目录

题目

[nuaa_question1](#)

[nuaa_question2](#)

[nuaa_question3](#)

[nuaa_question4](#)

[nuaa_question5](#)

[nuaa_question6](#)

题目

nuaa_question1

给予整数n，返回第n个字节内所有偶数位都为1的32位整数。

```
int nuaa_question1(int n) {  
    return 0x55 << (n << 3);  
}
```

直接移位0x55即可，因为偶数位移8的整数倍还是偶数位。

nuaa_question2

给予整数n和x，函数使x内的第n个字节中的每个位反转。

```
int nuaa_question2(int x, int n) {  
    return x ^ (0x000000ff << (n << 3));  
}
```

$0 \wedge a = a$, $1 \wedge a = \sim a$ (a为一个位)。

nuaa_question3

给予整数x，y和n，函数判断x的第n个字节和y的第n个字节是否相等，若相等返回0，不相等返回1。

```
int nuaa_question3(int x, int y, int n) {  
    int mark = (0x000000ff << (n << 3));  
  
    return !((x & mark) ^ (y & mark));  
}
```

先用掩码和x，y进行&运算取出对应的位，然后通过 $a \wedge b = 0 \Leftrightarrow a == b$ 得出结果。

nuaa_question4

给予整数x, y和n, 函数使用y的第n个字节与x的相应字节做与操作, 并返回结果。

```
int nuaa_question4(int x, int y, int n) {
    int mark = (0x000000ff << (n << 3));

    return x & ((y & mark) ^ ~mark);
}
```

先用掩码操作将y的第n个字节取出。然后通过 $0 \wedge a = a$, $1 \wedge a = \sim a$ (a为一个位) 将除第n个字节外的其他字节转换为0xff, 再进行&运算即可。

nuaa_question5

给予整数x, 计算非x (即!x), 但不允许使用 !运算符。

```
int nuaa_question5(int x) {
    return 1 & (~(x | (~x + 1)) >> 31);
}
```

对于C语言, 一般来说, 一个整数的取反加一的结果为它的相反数, 除了0和0x80000000 (32位中)。而其中只有0和它自己取反加一的结果符号位都为0。通过符号位判断出0这个值然后取反后和1进行&运算 (因为要求0返回1, 非0返回0)。

nuaa_question6

给予一个以无符号整数表示的浮点数uf (你可以认为uf具有浮点数的比特级结构), 函数返回2*uf (返回的结果也是一个无符号整数表示的浮点数); 如果uf为NaN, 请直接返回它本身。

```
unsigned nuaa_question6(unsigned uf) {
    unsigned ans = uf & 0x7f800000;

    if (ans != 0x7f800000 && ans != 0) {
        ans = uf + 0x00800000;
    } else if (!ans) {
        ans = ((uf & 0x007fffff) << 1) | (uf & 0x80000000);
    } else {
        ans = uf;
    }

    return ans;
}
```

首先把变量ans作为阶码使用, 判断阶码的位是否全为0或全为1。都不是就按正常情况处理, 将uf阶码加一并将结果赋值ans返回。全为0说明为非规格化数, 将尾数左移一位 (这里会自动进位), 保留符号位并将结果赋值ans返回。全为1说明为无穷大或者NaN, 直接返回本身。