

E3 - Solution

A Firefly小姐的位运算大练习—基础版

难度	考点
1	位运算

题目分析

在题目给定的模版中填充计算的部分即可。

示例代码

```
#include <stdio.h>

int main(){
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        int op;
        scanf("%d", &op);
        if (op == 1) {
            unsigned int a, b;
            scanf("%u %u", &a, &b);
            printf("%u\n", a & b);
        }else if (op == 2) {
            unsigned int a, b;
            scanf("%u %u", &a, &b);
            printf("%u\n", a | b);
        }else if (op == 3) {
            unsigned int a, b;
            scanf("%u %u", &a, &b);
            printf("%u\n", a ^ b);
        }
    }
}
```

Author : LiLi

B 二进制表示!

难度	考点
1~2	二进制表示

题目分析

如题所示，我们需要输出原码，反码和补码。

计算机中存储数字使用补码。

原码就是符号位加上真值的绝对值，即用第一位表示符号，其余位表示值。

反码用于简化负数的表示。**正数的反码与其原码相同，而负数的反码是其原码的非符号位取反。**

补码是计算机存储数字的形式。**正数的补码与其原码相同，而负数的补码是其反码加1。**

示例代码

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int t,a,b;
    int n;
    scanf("%d",&n);
    int m = abs(n);
    if(m==n){
        for(int i=31;i>=0;i--) printf("%d", (m>>i)&1);
        printf("\n");
        for(int i=31;i>=0;i--) printf("%d", (m>>i)&1);
        printf("\n");
        for(int i=31;i>=0;i--) printf("%d", (m>>i)&1);
        printf("\n");
    }
    else{
        printf("%d",1);
        for(int i=30;i>=0;i--) printf("%d", (m>>i)&1);
        printf("\n");
        printf("%d",1);
        for(int i=30;i>=0;i--) printf("%d", !((m>>i)&1));
        printf("\n");
        for(int i=31;i>=0;i--) printf("%d", ((n>>i)&1));
        printf("\n");
    }
    return 0;
}
```

C 我要吃到最后一块糖果！ (Easy Version)

难度	考点
2	位运算

题目分析

由于每次只能拿 2 的幂次个，所以可以把糖果总数 n 考虑成二进制，因为需要尽可能得多拿，所以，可以理解成：每一次需要把总数 n 的二进制最高位变成 0，考虑一共可以变化几次即可。例如给出的第一个样例：

$$(31)_{10} = (11111)_2$$

每次拿走最高位的 1，一共可以拿 5 次。注意到，5 是奇数，所以 Alice 拿走最后一块。

所以本题只需判断 n 的二进制表示中 1 的数量的奇偶，进一步分析，只需判断出 n 的二进制表示中各位数字之和的奇偶即可。

示例代码

```
#include <stdio.h>
int main(){
    int t;
    scanf("%d",&t);
    while(t--){
        unsigned int n;
        scanf("%u",&n);
        int k=0;
        while(n){
            k=k+(n&1);
            n=n>>1;
        }
        if(k&1){
            printf("Alice\n");
        }
        else{
            printf("Bob\n");    //千万不要忘记输出换行!!!!!!
        }
    }
    return 0;
}
```

D 蒙版工具

难度	考点
3	位运算

题目分析

按照题意作答即可。

示例代码 - 1

```
#include <stdio.h>

int main() {
    unsigned int x, y, mask;
    while(scanf("%u%u%u", &x, &y, &mask) != EOF) {
        unsigned int res = 0;
        for(int i = 0; i < 32; ++i) {
            int bit = (mask >> i) & 1;
            if(bit == 1) {
                int bit_y = (y >> i) & 1;
                if(bit_y == 1) {
                    res |= 1 << i;
                }
            }
            else {
                int bit_x = (x >> i) & 1;
                if(bit_x == 1) {
                    res |= 1 << i;
                }
            }
        }
        printf("%u\n", res);
    }
    return 0;
}
```

示例代码 - 2

```
#include <stdio.h>

int main() {
    unsigned int x, y, mask;
    while(scanf("%u%u%u", &x, &y, &mask) != EOF) {
        unsigned int res = (x&(~mask)|(y&(mask)));
        printf("%u\n", res);
    }
    return 0;
}
```

E Firefly小姐的位运算大练习—进阶版

难度	考点
3	位运算

题目分析

在题目给定的模版中填充计算的部分即可。

本题中的位运算操作都是考试常见操作，需要熟练掌握。

示例代码

```
#include <stdio.h>

int main(){
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        int op;
        scanf("%d", &op);
        if (op == 1) { // 二进制位1的个数
            unsigned int a;
            scanf("%u", &a);
            int res = 0; // 初始化
            for (int i = 0; i < 32; i++) {
                if ((a >> i & 1) == 1) { // 将a的第i位通过右移移至第一位，再&1将其余部分
置0
                    res++; // 统计
                }
            }
            printf("%d\n", res);
        }else if (op == 2) {
            unsigned int a;
            int l;
            scanf("%u %d", &a, &l);
            printf("%u\n", a ^ (1 << l)); // ^1翻转，^0不变，^(1 << l)可以达到其余位不
变，l位翻转的目的
        }else if (op == 3) {
            unsigned int a;
            int l;
            scanf("%u %d", &a, &l);
            printf("%u\n", a & ~(1 << l)); // &1不变，&0置0，&~(1 << l)可以达到其余位
不变，l位置0的目的
        }else if (op == 4) {
            unsigned int a;
            int l;
            scanf("%u %d", &a, &l);
            printf("%u\n", a | (1 << l)); // |1置1，|0不变，|(1 << l)可以达到其余位不
变，l位置1的目的
        }else if (op == 5) {
            unsigned int a;
            int l;
            scanf("%u %d", &a, &l);
            printf("%u\n", a >> l & 1); // 将a的第l位通过右移移至第一位，再&1将其余部分
置0
        }
    }
}
```

Author : LiLi

F Firefly小姐的位运算大练习—最终版

难度	考点
3	位运算

题目分析

在题目给定的模版中填充计算的部分即可。

本题中的位运算操作都是考试常见操作，需要熟练掌握。

示例代码

```
#include <stdio.h>

int main(){
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        int op;
        scanf("%d", &op);
        if (op == 1) {
            unsigned int a;
            int l, r;
            scanf("%u %d %d", &a, &l, &r);
            for (int i = l; i <= r; i++) { // 循环l至r位的单位翻转操作（参考E），下同
                a ^= 1 << i;
            }
            printf("%u\n", a);
        }else if (op == 2) {
            unsigned int a;
            int l, r;
            scanf("%u %d %d", &a, &l, &r);
            for (int i = l; i <= r; i++) {
                a &= ~(1 << i);
            }
            printf("%u\n", a);
        }else if (op == 3) {
            unsigned int a;
            int l, r;
            scanf("%u %d %d", &a, &l, &r);
            for (int i = l; i <= r; i++) {
                a |= 1 << i;
            }
            printf("%u\n", a);
        }else if (op == 4) {
            unsigned int a, b;
            int l, r;
            scanf("%u %u %d %d", &a, &b, &l, &r);
```

```

        for (int i = 1; i <= r; i++) {
            a &= ~(1 << i); // 将a的第i位空出来
            a |= (b & (1 << i)); // 取出b的第i位，接到a上
        }
        printf("%u\n", a);
    }else if (op == 5) {
        unsigned int a;
        int l, r;
        scanf("%u %d %d", &a, &l, &r);
        unsigned x = a >> l & 1, y = a >> r & 1; // 取出a的l和r位
        a &= ~(1 << l); // 空出a的第l位
        a &= ~(1 << r);
        a |= x << r;
        a |= y << l;
        printf("%u\n", a);
    }
}
}

```

Author : PiPi

G s7h的简单异或题

难度	考点
2	位运算

题目分析

本题给定一个数列，多次询问求 l 到 r 的异或和。

首先根据异或的运算律可以推导出一个性质：

$$(a_1 \oplus a_2 \oplus \dots \oplus a_r) \oplus (a_1 \oplus \dots \oplus a_{l-1}) = a_l \oplus a_{l+1} \dots \oplus a_r。$$

然后类比 E2J，构建一种类似前缀和的“前缀异或和”算法即可。

代码

```

#include<stdio.h>
int n,m,a[200050],l,r,sum[200050];
int main(){
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;++i){
        scanf("%d",&a[i]);
        sum[i]=a[i]^sum[i-1];
    }
    for(int i=1;i<=m;++i){
        scanf("%d%d",&l,&r);
        printf("%d\n",sum[r]^sum[l-1]);
    }
    return 0;
}

```

H 小霁的时间乱流plus+++

难度	考点
4	数组 循环

题目分析

本题在分钟进位小时，小时进位天数较为简单。

但在天数进位月份时，需要考虑每月天数不同也需要考虑闰年的因素。

闰年的规定是“能被400整除，或者能被4整除但不能被100整除的是闰年”，转化为C语言就是 `((y % 4 == 0 && y % 100 != 0) || y % 400 == 0)`，如果该条件式成立，则y是闰年，反之不是闰年。具体的乱流中的时间调整方式已在题目中详细给出，此处不再赘述。

示例代码

```
#include<stdio.h>
int main()
{
    int months[13]={0,31,28,31,30,31,30,31,31,30,31,30,31}; //定义每月天数
    int n; //数据组数
    int y,mon,d,h,min; //年、月、日、时、分
    int tempday;
    scanf("%d",&n);
    for (int i = 0; i < n; ++i) {
        scanf("%d.%d.%d %d:%d",&y,&mon,&d,&h,&min); //输入乱流中的时间
        h+=(min/60);
        min%=60;
        d+=(h/24);
        h%=24; //计算正常的小时数、分钟数，并向天数进位
        y += (mon - 1) / 12; mon = (mon - 1) % 12 + 1; //先将月份向年份进位
        tempday=months[mon]+(((y % 4 == 0 && y % 100 != 0) || y % 400 == 0)&&(mon==2)); //计算当前的y和mon对应的当月天数
        //(((y % 4 == 0 && y % 100 != 0) || y % 400 == 0)&&(mon==2))用于判定当前的y和mon是否是闰年的2月，若条件式成立，则条件式的值为1，在28天的基础上加1天。
        while (d>tempday) //如果天数超过该月份的天数
        {
            d-=tempday;
            mon++; //天数减去该月天数，月份加一
            y += (mon - 1) / 12; mon = (mon - 1) % 12 + 1; //若月份大于12，则向年份进位
            tempday=months[mon]+(((y % 4 == 0 && y % 100 != 0) || y % 400 == 0)&&(mon==2)); //计算当前的y和mon对应的当月天数
        }
        printf("%04d.%02d.%02d %02d:%02d\n",y,mon,d,h,min);
    }
}
```


I -M3rcury-的解方程组

难度	考点
5	位运算

题目分析

我们观察异或运算对应的真值表：

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

可以发现，在二进制对应的位上，异或运算和加法运算的结果一样，只是忽略了进位。

那么，对于 $x + y$ 和 $x \oplus y$ 来说，他们的差值就是中间产生进位的值。

只有 x 和 y 对应位都为 1 时，才会在高一位的地方产生进位。

因此我们可以得到

$$x + y = (x \oplus y) + 2 * (x \& y)$$

所以

$$x \& y = \frac{(x+y) - (x \oplus y)}{2}$$

由于 $x \geq x \& y$ ，因此 $x \& y$ 即为 x 的最小值， $a - x$ 即为对应的 y

最后考虑无解的情况。

因为这样求出的 y 不一定满足 $x \oplus y = b$ ，因此需要代回去验证

此外 $a < b$ 的情况也是无解的

示例代码

```
#include <stdio.h>
int main()
{
    unsigned long long a,b;
    scanf("%llu%llu",&a,&b);
    if(a<b)
    {
        printf("-1\n");
    }
    else
    {
        unsigned long long x=a-b>>1;
        unsigned long long y=a-x;
```

```

        if((x^y)!=b) printf("-1\n");
        else printf("%11u %11u\n",x,y);
    }
    return 0;
}

```

J IEEE 754科学

难度	考点
5	数据类型

题目分析

本题分为两部分：构造浮点数和输出浮点数的二进制表示。

[illegible][illegible]

拷贝方法如下：

```
unsigned long long u = 0x7effffffffffffffff; // u = 1
double a;
memcpy(&a, &u, sizeof(double)); // u拷贝到a
```

对于输出浮点数的二进制表示，读入后参考 Hint 拷贝，然后位运算逐位输出即可，不再赘述。

示例代码 - 1

```
#include<stdio.h>
#include<string.h>
int main() {
    unsigned long long d;
    double a;
    d = 0x7FEFFFFFFFFFFFFFFF;
    memcpy(&a, &d, sizeof(double));
    printf("%.2f\n", a);
    d = 1;
    memcpy(&a, &d, sizeof(double));
    printf("%.1100f\n", a);
    int m;
    scanf("%d", &m);
    for (int i = 0; i < m; i++) {
```

```

scanf("%lf", &a);
memcpy(&d, &a, sizeof(double));
for (int j = 63; j >= 0; j--) {
    printf("%llu", (d >> j) & 1);
}
printf("\n");
}
return 0;
}

```

示例代码 - 2

推荐大家学了指针以后再来体会这份代码。

```

#include<stdio.h>

int main() {
    unsigned long long d;
    d = 0x7FEFFFFFFFFFFFFFFF;
    printf("%.2f\n", *(double *)&d);
    d = 1;
    printf("%.1100f\n", *(double *)&d);
    int m;
    scanf("%d", &m);
    for (int i = 0; i < m; i++) {
        scanf("%lf", (double *)&d);
        for (int j = 63; j >= 0; j--) {
            printf("%llu", (d >> j) & 1);
        }
        printf("\n");
    }
    return 0;
}

```

示例代码 - 3

推荐大家学了联合体（不是结构体）以后再来体会这份代码。

```

#include<stdio.h>
union{
    double a;
    unsigned long long d;
}num;
int main() {
    num.d = 0x7FEFFFFFFFFFFFFFFF;
    printf("%.2f\n", num.a);
    num.d = 1;
    printf("%.1100f\n", num.a);
    int m;
    scanf("%d", &m);
    for (int i = 0; i < m; i++) {
        scanf("%lf", &num.a);
        for (int j = 63; j >= 0; j--) {
            printf("%llu", ((num.d) >> j) & 1);
        }
    }
}

```

```

    }
    printf("\n");
}
return 0;
}

```

K Firefly!!!

难度	考点
3	空白字符，字符串匹配

题目分析

本题的考点在于利用空白字符（'\n', EOF）判断输入的字符画和标准字符画是否相同。

我们可以先统计第一个空行出现前读入的 f 个数确定需要匹配的标准 F 型字符画尺寸，再一一进行匹配。

示例代码

```

#include <stdio.h>

int main() {
    int n = 0;
    while (getchar() != '\n') { // F型字符画尺寸
        n++;
    }
    for (int i = 0; i < n - 2; i++) { // 判断接下来是否是连续n-2行每行一个字符f（即连续读入n-2个'f'和'\n'）
        char a, b;
        a = getchar();
        b = getchar();
        if (!(a == 'f' && b == '\n')) { // 匹配失败，输出No并使用return 0直接终止程序
            printf("No");
            return 0;
        }
    }
    for (int i = 0; i < n; i++) { // 判断接下来是否是连续n个字符f
        char a;
        a = getchar();
        if (a != 'f') {
            printf("No");
            return 0;
        }
    }
    for (int i = 0; i < n - 1; i++) { // 判断接下来是否是连续n-1行每行一个字符f（即连续读入n-1个'\n'和'f'）
        char a, b;
        a = getchar();
        b = getchar();
        if (!(a == '\n' && b == 'f')) {

```

```
        printf("No");
        return 0;
    }
}
char c;
c = getchar();
if (c == EOF) { // 判断是否已经读到字符画末尾，即后面没有更多字符
    printf("Yes %d", n);
}else
    printf("No");
}
```

Author : LiLi