

A> HugeGun学姐的巧克力棍

三角形的判断没有什么特别困难的地方，但是可以借助else来简化自己的判断。

这个题顺便也考察了转义字符，需要注意的是%转义是使用%即printf("%%");

输出的时候一定要借助样例，要看仔细。

```
#include<stdio.h>

int main()
{
    int a, b, c;
    scanf("%d%d%d", &a, &b, &c);
    if (a + c > b && a + b > c && b + c > a) {
        if (a == b || a == c || b == c) {
            if (a == b && b == c) {
                printf("equilateral triangle!\\");
            } else {
                printf("isosceles triangle!\\\\\\");
            }
        } else {
            printf("regular triangle!%%d");
        }
    } else {
        printf("not a triangle!\\n");
    }
    return 0;
}
```

B> 中位数计算

根据逻辑运算的结果为真其实是用整型数字1来表示，可以快速求出中位数。

```
#include <stdio.h>

int main() {
    int a[5];
    int i, j, temp;
    for (i = 0; i < 5; i++)
        scanf("%d", &a[i]);
    int sum = 0;
    for (i = 0; i < 5; i++) {
        sum += a[i];
        if ((a[i] < a[0]) + (a[i] < a[1]) + (a[i] < a[2]) +
            (a[i] < a[3]) + (a[i] < a[4]) == 2) {
            printf("%d\\n", a[i]);
        }
    }
}
```

```

    }
}
printf("%.2f\n", (double)sum / 5);
return 0;
}

```

C> 表

首先，每12小时一循环，先将读入的第 x 小时内的 x 对12取模得到输出的第一个数，即表盘上的小时数 x 。设现在为 x 点且 y 分钟后，时针和分针重合。一分钟时针走 $\frac{30}{60}$ 度，分针走 $\frac{360}{60}=6$ 度，可列出方程： $30x + \frac{1}{2}y = 6y$ 即 $y = \frac{60x}{11}$ ，最后再特判一下 $y=60$ 和 $x=12$ 的情况后把 y 的小数部分转为秒数即可。

```

#include<stdio.h>
#include<math.h>
const double eps = 1e-8;
int main() {
    int x, yy;
    double y;
    scanf("%d", &x);
    x %= 12;
    y = x * 60. / 11;
    if (fabs(y - 60.) < eps) {
        x++;
        y = 0;
    }
    yy = y;
    if (x == 12) x = 0;
    printf("%d:%d:%.7f", x, yy, (y - yy) * 60);
    return 0;
}

```

D> 矩形面积

矩阵覆盖的面积就是两个矩阵的面积之和减去两个矩阵相交的面积。

重点就在于两个矩阵相交的面积。

矩形面积就是底乘高，我们只需要算底和高的长度即可。

这样就转换成了求两条线段相交的长度。

具体算法看代码注释。

```

#include <stdio.h>

```

```

//求两条线段相交的长度
//ab, cd分别为两条线段
int ins(int a, int b, int c, int d)
{
    //假设存在相交线段, 依次计算相交线段最左和最右的点
    c = a < c ? c : a;
    b = b < d ? b : d;
    //注意, 如果不相交, 应该返回0, 而不是负值
    return b > c ? b - c : 0;
}

//计算覆盖面积
int computeArea(int a, int b, int c, int d, int e, int f, int g, int h)
{
    int ins1 = ins(a, c, e, g);
    int ins2 = ins(b, d, f, h);
    //矩阵面积之和减去相交面积
    return (c - a) * (d - b) - ins1 * ins2 + (g - e) * (h - f);
}

int main()
{
    int a, b, c, d, e, f, g, h;
    scanf("%d%d%d%d%d%d%d", &a, &b, &c, &d, &e, &f, &g, &h);
    printf("%d\n", computeArea(a, b, c, d, e, f, g, h));
    return 0;
}

```

E> 原码和补码

见代码

```

#include <stdio.h>
#include <math.h>
#include <time.h>
#include <stdlib.h>

int main()
{
    long long a = 1 << 30;
    int b, i;
    a <= 1;
    for (i = 6; i <= 10; ++i)
    {
        scanf("%d", &b);

        while (a)
        {
            if (a & b)
            {

```

```
                putchar('1');
            }
            else
            {
                putchar('0');
            }
            a = a >> 1;
        }
    }
}
```

F> login学数论

本题需要寻找[n,m]之间的质数。

针对单个数i，可以通过判断是否存在小于j小于i，使得j|i。如果不存在，则说明i是质数。

因此，使用两个for循环嵌套即可。

```
#include<stdio.h>

int main() {
    int n,m;
    scanf("[%d,%d]", &n, &m);
    int i;
    for (i = n; i <= m; i++) {
        int is_prime = 1;
        int j;
        for (j = 2; j < i; j++) {
            if(i % j == 0) {
                is_prime = 0;
                break;
            }
        }
        if (is_prime) {
            printf("%d ",i);
        }
    }
    return 0;
}
```