

## 问题 A: 探基不识火柴棒

本题取出输入整数的每一位数，然后通过多分支判断使用的火柴棒数。

在通过循环获取每一位数时，要考虑到输入数为 0 的情况。

如果使用数组，记录每个月使用的火柴棒数，程序会更简单。

```
int main()
{
    int x,c=0;
    int arr[10]={6,2,5,5,4,5,6,3,7,6};
    scanf("%d",&x);
    do
    {
        c+=arr[x%10];
        x=x/10;
    }while(x>0);
    printf("%d\n",c);
}
```

## 问题 B: 探姬同学的名字

签到题，两个数乘积。

## 问题 C: 探姬的晨跑计划

本题可以通过循环判断每一天跑的圈数，累加起来。对于每月天数，也就是循环的次数，可以通过多分支结构判断，也可以通过打表的方式，用数组记录每个月的天数。

```
int main() {
    int i, m, w, odd, even, d, count;
    int arr[7] = {0}, month[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    scanf("%d%d", &m, &w);
    scanf("%d%d", &odd, &even);
    arr[1] = arr[3] = arr[5] = odd;
    arr[2] = arr[4] = arr[6] = even;
    count = 0;
    for (i = 1; i <= month[m]; i++) {
        count += arr[w];
        w = (w + 1) % 7;
    }
    printf("%d\n", count);
    return 0;
}
```

## 问题 D: 探姬的购物车

本题是循环累加问题，需要注意以下几个问题，输入商品单价要用 `double` 类型保存，没有给定商品数量，此类问题在循环输入时，要用 `EOF`，`while(scanf("%lf %d",&price,&c)!=EOF)` 或 `while(~scanf("%lf %d",&price,&c))`，在选择优惠方式时，可以同时用两种优惠，也可以只使用一种优惠方式，有时候使用两种优惠不如使用一种满减优惠，所以要比较一下哪种优惠幅度更大再输出。

```

int main()
{
    int m, s, c;
    double a1, a2, discount, price, total;
    scanf("%d %d", &m, &s);
    scanf("%lf", &discount);
    while (~scanf("%lf %d", &price, &c))
        total += price * c;
    a1 = total - (int)(total / m) * s;
    a2 = total * discount - (int)(total * discount / m) * s;
    printf("%.2f", a1 > a2 ? a2 : a1);
    return 0;
}

```

## 问题 E: 探姬同学去上课

本题是一简单循环结构，在进行循环时，注意高于探姬的楼层按键无效。

```

int main()
{
    int i, tj, n, k, sum;
    scanf("%d", &tj);
    scanf("%d", &n);
    sum = (f-1)*2; //中间不停时的总用时
    while (n--)
    {
        scanf("%d", &k);
        if (k < tj)
            sum += 3; //有开关电梯时间再加3，前面每层用时都按2算的。
    }
    printf("%d\n", sum);
    return 0;
}

```

## 问题 F: 探姬帮助同学解决问题

字母、数字、空格、标点符号的 ASCII 码值，均大于 20，小于 124，对于输入的数字字符串，我们可以一个字符一个字符的读取，也可以一位一位整数读取，在循环读取的过程中判断读取的数是否大于 20，如果大于 20 直接输出对应的字符，并把保存读取数值的变量置为 0。

```

int main()
{
    int a, i = 0;
    while (~scanf("%1d", &a))
    {
        i = i * 10 + a;
        if (i > 20)
        {
            printf("%c", i);
            i = 0;
        }
    }
    return 0;
}

```

## 问题 G: 探姬环游世界?

简单循环，标志变量的使用，用  $f$  记录读取过的数是否为 0。通过循环读取每一个数，如果不为 0，把标志变量置为 0，如果读取到的数是 0，通过标志变量判断之前读取到的数是否为 0，如果  $f==0$ ，则之前读取到的不为 0，换女装计数加 1。

```
int main()
{
    int a,n,i=0,c=0,f=0; //f用来记录读过的数是否为0
    scanf("%d",&n);
    while(n-->0)
    {
        scanf("%d",&a);
        if(a!=0)
            f=0;
        else if(f==0)
        {
            c++;
            f=1;
        }
    }
    printf("%d",c);
    return 0;
}
```

## 问题 H: 探姬粉丝众多

本题为一分支结构，主要判断一下什么情况下能够推算出总人数，要推算出总人数需要知道列数或者有一组输入的位置在第一行，要推算出列数，需要两组输入的位置不在同一行。

```
int main(){
    int r1,c1,n1,r2,c2,n2,cols;
    scanf("%d%d%d",&n1,&r1,&c1);
    scanf("%d%d%d",&n2,&r2,&c2);

    if(r1==r2 && r1>1)
        printf("sorry!\n");
    else if(r1*r2==1)
        printf("%d\n",c1+n1-1);
    else
    {
        cols=(n1+c1-n2-c2)/(r2-r1);
        printf("%d\n",(r1-1)*cols+n1+c1-1);
    }
}
```

## 问题 I: 探姬给同学们送糖果

本题为防学长 AK 题，我们可以探姬同学的位置为中心，一圈一圈往外推算，每一圈所有位置的距离是一样的，在推算的过程中，可能会遇到边界，可以用 a,b,c,d 分别表示每一圈上左上角、右上角、右下角、左下角的四边有多少人，其中左上角的边包含上顶点，右上角的边包含右顶点。

```
int n, m, sx, sy;
int min(int x, int y)
{
    return x > y ? y : x;
}
int max(int x, int y)
{
    return x > y ? x : y;
}
int calc(int k) {
    int r = min(sx + sy - k - 1, sx - 1);
    int l = max(1, sx - k);
    int a = max(0, r - l + 1);
    r = min(sx - k - sy + m, sx);
    l = max(1, sx - k + 1);
    int b = max(0, r - l + 1);
    r = min(sx + k, n);
    l = max(sx + 1, sx + sy + k - m);
    int c = max(0, r - l + 1);
    r = min(sx + k - 1, n);
    l = max(sx, sx - sy + k + 1);
    int d = max(0, r - l + 1);
    return a + b + c + d;
}
int main() {
    int k; scanf("%d%d%d%d", &n, &m, &sx, &sy, &k);
    long long ans1 = 0;
    for (int i = 1; k > 0; i += 2) {
        int w = calc(i);
        if (w > k) { ans1 += 111 * i * k; break; }
        ans1 += 111 * i * w; k -= w;
    }
    printf("%11d\n", ans1 * 2);
    return 0;
}
```

## 问题 J: 探姬同学@出题人

本题为输出图型题，通过双层循环判断每一个位置是空格还是@字符，通过多分支对首末行，中间行和其它行进行分别判断，在其它行判断时，还要考虑是否为第二行或倒数第二行。

```
int main() {
    int i, j, n;
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        for (j = 0; j < n * 3 + 4; j++) {
            if (i == 0 || i == n - 1) {
                if (j < n || j == n + 2 || j == 2 * n + 1 || j > 2 * n + 3 && j < n * 3 + 2)
                    printf("@");
                else
                    printf(" ");
            } else if (i == n / 2) {
                if (j == n / 2 || j > n + 1 && j <= 2 * n + 1 || j == 2 * n + 4 || j == 3 * n + 3)
                    printf("@");
                else
                    printf(" ");
            } else {
                if (j == n - 1 - i || j == n + 2 || j == 2 * n + 1 || j == 2 * n + 4 || (i == 1 || i == n - 2) ? (j == 3 * n + 2) : (j == 3 * n + 3))
                    printf("@");
                else
                    printf(" ");
            }
        }
        printf("\n");
    }
    return 0;
}
```