

每次输入一个算式，对运算符采用 `switch` 判断，分别输出不同的运算结果。（写成级联的if-else分支判断也可以）。然后注意输出的格式，对于除法要保留两位小数并且需注意类型转换 `1.0*a/b` 。

示例代码

```
#include <stdio.h>
int main()
{
    int a, b;
    char c;
    while (scanf("%d %c %d", &a, &c, &b) != EOF) {
        switch (c) {
            case '+':
                printf("%d\n", a + b);
                break;
            case '-':
                printf("%d\n", a - b);
                break;
            case '*':
                printf("%d\n", a * b);
                break;
            case '/':
                if (b == 0)
                    puts("Runtime Error(SIGFPE)");
                else
                    printf("%.2f\n", 1.0 * a / b);
                break;
            case '%':
                if (b == 0)
                    puts("Runtime Error(SIGFPE)");
                else
                    printf("%d\n", a % b);
                break;
        }
    }
    return 0;
}
```

c 我的萌王不可能这么可爱

难度	考点
1	循环

题目分析

无难点，签到题。

示例代码

```
#include<stdio.h>
int main()
{
    int i,a,s,n;
    while(scanf("%d",&n)!=EOF)
    {
        for(i=1,s=0;i<=n;++i)
            scanf("%d",&a),s+=(a==i);
        printf("%d\n",s);
    }
    return 0;
}
```

D 数字反转

难度	考点
2	字符串、循环

题目分析

使用字符串读入，按要求除去前导0后倒序输出即可，注意只去掉前导0

示例代码

```
#include<stdio.h>
int main(){
    char m[3070];
    char n[3070];
    gets(m);
    int l=strlen(m);
    int j=l-1;
    for(int i=0;i<l;i++)
        n[j--]=m[i];
    int i=0;
    while(n[i] == '0' && i!=l-1)
        i++;
    while(i<l)
        printf("%c",n[i++]);
    return 0;
}
```

E jqe的整数1

难度	考点
2	位运算

题目分析

本题考察位运算，使用 & 运算可以取出需要的数据，再使用位运算将数据移动到需要的位置。使用加法将每个位置的数据加在一起。

一定要使用 unsigned int 或 long long ，否则会出现负数的问题。

示例代码

```
#include <stdio.h>

int main()
{
    unsigned int a, b;
    scanf("%u", &a);
    b = ((a & 0xFF000000U) >> 24) +
        ((a & 0X00FF0000U) >> 8) +
        ((a & 0X0000FF00U) << 8) +
        ((a & 0X000000FFU) << 24);
    printf("%u", b);
    return 0;
}
```

F 斯波利特平衡术

难度	考点
3	数组、循环

题目分析

本题考察一维数组的运用与循环的使用。

这里给出一种反转区间 $[l,r]$ 的伪代码，这种写法的 while 最多循环 n 次（想一想，为什么）：

```
while(l < r){
    交换 a[l], a[r];
    l = l + 1;
    r = r - 1;
}
```

~~当然也可能有同学看出这是一道平衡树板子题，然后秒切子~~

示例代码

```

#include <stdio.h>

#define N (1000 + 5)

int a[N];

int main(){
    int q;
    int n, m;
    int l, r;
    int i, t;

    scanf("%d", &q);
    while (q--){
        scanf("%d%d", &n, &m);
        for (i = 1; i <= n; i++){
            a[i] = i;
            while (m--){
                scanf("%d%d", &l, &r);
                while (l < r){
                    t = a[l];
                    a[l] = a[r];
                    a[r] = t;
                    l++;
                    r--;
                }
            }
        }
        for (i = 1; i <= n; i++){
            printf("%d ", a[i]);
        }
        printf("\n");

        return 0;
    }
}

```

G 粉丝的秘密

难度	考点
3	循环

题目分析

模拟每一个人的小秘密的传递过程，传递路径是唯一的，且成功传递到最后一个位置的最大次数为 $n-1$ 次，所以模拟完所有小秘密的传递过程就可以得到结果；

示例代码

```
#include <stdio.h>
int main()
{
    int n;
    int a[1005] = {0};
    int change, count, point, num;
    int i;

    while (scanf("%d", &n) != EOF)
    {
        change = 0;
        count = 0;
        for (i = 1; i <= n; i++)
            scanf("%d", &a[i]); //读入数据

        for (i = 1; i < n; i++) //对每个秘密的传递过程进行遍历
        {
            point = i;
            num = 0;
            do
            {
                num++;
                point = a[point];
                if (num >= n + 2) //传递如果超过n+2次，说明永远不会传到最后一个人
                {
                    printf("No solution!\n");
                    change = 1;
                    break;
                }
            } while (point != n); //可以传递到最后一个人

            if (change)
                break;
            if (num > count)
                count = num;
        }
        if (change == 0) //有效的传递
            printf("%d\n", count);
    }
    return 0;
}
```

H 猪国杀蚂蚁

难度	考点
4	哈希表

题目分析

用 $10^5 + 1$ 长度的数组 a 存储每个编号出现的次数,即读入到 t 则将 $a[t] + 1$, 由于题目保证做过同一题的同学编号不会重复, 可以遍历 1 到 10^5 , 若 $a[i] == 1$, 则输出 i , 无需额外判断.

注意每组数据要将数组 a 初始化为 0

示例代码

```
#include<stdio.h>

int n,m,a[111111],t;

int main()
{
    int i;
    while(scanf("%d%d",&n,&m) > 0)
    {
        for(i = 1;i <= 100000;i++)
            a[i] = 0;
        for(i = 1;i <= n + m;i++)
        {
            scanf("%d",&t);
            a[t]++;
        }
        for(i = 1;i <= 100000;i++)
            if(a[i] == 1)
                printf("%d ",i);
        printf("\n");
    }

    return 0;
}
```

I 硬币的组合

难度	考点
5	递归、递推

解题思路

递推思路：这是个完全背包问题（详细点链接跳转）。

https://blog.csdn.net/qq_38984851/article/details/81133840

递归思路：对于*i*元资产的家庭，如果*i*为奇数，则组合数与*i-1*元资产家庭的组合数相同（即每种方案多加一元硬币）。如果*i*为偶数，则需要在*i-1*元资产的方案上，加上不包括1元硬币的方案数，也就是*i/2*元资产的方案数（如果都是2元以上，每一组除以2就是*i/2*资产的方案）。即有公式：

$f(i)=f(i-1)+(i\%2==0?f(i/2):0)$

示例程序

```
#include<stdio.h>    //背包
int f[10][205];
int main()
{
    int n,c=0;
    scanf("%d",&n);
    for (int j=0;j<=n;j++) f[0][j]=1;
    for (int i=1;i<=8;i++)
    {
        if ((1<<i) >n) break;
        c=i;
        for (int j=1;j<=n;j++)
        {
            for (int k=0;k<=n;k++)
            {
                if ((1<<i)*k >j) break;
                f[i][j]+=f[i-1][j-(1<<i)*k];
            }
        }
    }
}
```

```
#include<stdio.h> //递归

int main(){
    int i,n,b[202]={1,1,2};
    for(i=3;i<202;i++){
        b[i]+=b[i-1];
        if(i%2==0)
            b[i]+=b[i/2];
    }
    scanf("%d",&n);
    printf("%d\n",b[n]);
    return 0;
}
```

K 回文质数商店

难度	考点
2	循环，数学

题目分析

这是一个需要循环的数学题，示例中使用函数可以使程序可读性更高。

判断一个数是否是回文数可以将一个数的各个位拆开；

判断一个数是否是质数只需要循环2到根号a，由于浮点数计算有精度和速度问题，我们使用乘方代替开根号。

每次读入一个数，查找下一个会打折的数，对比目前的价格，输出最少花费。

示例代码

```

#include <stdio.h>
#include <math.h>
int is_Pa(int a)
{ //判断是否是回文
    int b = 0, c = a;
    while (c)
    {
        b = b * 10 + c % 10;
        c /= 10;
    }
    if (a == b)
        return 1;
    else
        return 0;
}
int is_Prime(int n)
{ //判断是否是质数
    int i;
    if (n < 2)
    {
        return 0;
    }
    for (i = 2; i * i <= n; ++i)
    {
        if (n % i == 0)
        {
            return 0;
        }
    }
    return 1;
}
int main()
{
    int x, i, nxt = 0;
    scanf("%d", &x);
    for (i = x; i <= 1003001; i++)
        if (is_Prime(i) && is_Pa(i))
        {
            nxt = i;
            break;
        }
    if (x < (nxt + 1) / 2)
        printf("%d", x);
    else
        printf("%d", (nxt + 1) / 2);
    return 0;
}

```