

A.飞翔的天使（简单版）

题目描述

天使Cindy非常喜欢在空中翱翔。

有一天，天使想去找Potassium。我们只观察天使的飞行轨迹投影在地面上的部分。也即，可以用二维坐标系表示她的飞行状态。

严格来讲，设天使的速度为 (v_x, v_y) ，必须满足 $0 \leq v_x^2 + v_y^2 \leq v_{max}^2$ 。

初始时，她在起点 (s_x, s_y) ，她的终点是 (t_x, t_y) 。天使会选择最快到达终点的策略。Potassium想要在天使到达终点前在终点等她，问他最晚什么时候到达终点才能保证比天使更早到达？

输入

多组数据。

对于每一组数据，一行五个整数 s_x, s_y, t_x, t_y, v ，分别表示起点和终点坐标，以及她的最大速度。

保证所有输入数字的绝对值在50000内。

输出

对于每一组数据，输出一行一个浮点数，表示到达终点的最早时间 T 。当输出与正解的绝对误差不超过 10^{-5} 时，判断为正确。

输入样例

```
1 | 0 0 3 4 1
2 | 3 4 0 0 1
```

输出样例

```
1 | 5
2 | 5
```

样例解释

两组数据中，天使均可以用最快的速率在第5个单位时间到达终点。

hint

`int`能够表示的数范围有限（ $-2147483648 \sim 2147483647$ ），当要求一个比较大的`int`类型数的平方的时候有以下两种选择：

1.利用`<math.h>`中的函数`pow`进行运算（`pow(x, 2)`），将 x 看做浮点数，计算结果。

2.类型转换成`long long`再进行运算（`1LL*x*x`或`(long long)x*x`）。

这里两种方法均可，但建议使用第二种方法，能不用浮点数的时候尽量避免使用浮点数，以免产生误差。

解题思路

毫无疑问，天使沿着直线飞行是最快的。

AC代码

```
1  #include<stdio.h>
2  #include<math.h>
3  int main(){
4      long long x1,y1,x2,y2,v;
5      double d;
6      while(scanf("%lld%lld%lld%lld",&x1,&y1,&x2,&y2,&v)!=EOF){
7          d=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
8          printf("%.5f\n",d/v);
9      }
10     return 0;
11 }
```

B.Zyy学姐的生日礼物5

题目描述

MountVoom送了Zyy学姐一只烤鸡作为她的生日礼物。

Zyy吃烤鸡有五种特别的配料，每种配料可以放1~6克，任意烤鸡的美味度是所有配料质量之和。

Zyy学姐今天想吃美味度为 x ($5 \leq x \leq 30$)的烤鸡，她想知道有哪些加配料方案。

Zyy学姐希望你将方案组成一个五位数以后输出。

即如果当前方案为2,3,3,2,1，你需要输出23321。

输入

一行一个数 x ，意义如题面描述。

输出

第一行至结束，每行一个数，按Zyy学姐的要求输出方案。

要求输出的五位数是从大到小的，这样Zyy学姐检查你的答案会很方便。

输入样例

7

输出样例

31111

22111

21211

21121

21112

13111

12211

12121

12112

11311

11221

11212

11131

11122

解题思路

本题主要考察大家对于循环的使用。注意到本题的数据量并不大(10^5 量级)，因此直接暴力循环就可以。一种方法是对每一种配料都从6到1（保证从大到小）进行循环，一共嵌套6层，可以使用for也可以使用while；另一种思路则是直接从大到小遍历所有的5位数，只要它的各位满足题目所述条件就输出即可。

AC代码1

```

1  #include<stdio.h>
2  int main()
3  {
4      int n;
5      scanf("%d",&n);
6      for(int a=6;a>=1;a--)
7      {
8          for(int b=6;b>=1;b--)
9          {
10             for(int c=6;c>=1;c--)
11             {
12                 for(int d=6;d>=1;d--)
13                 {
14                     for(int e=6;e>=1;e--)
15                     {
16                         if(a+b+c+d+e==n)
17                         {
18                             printf("%d%d%d%d%d\n",a,b,c,d,e);
19                         }
20                     }
21                 }
22             }
23         }
24     }
25     return 0;
26 }
```

AC代码2

```

1  #include<stdio.h>
2  int main(){
3      int x,n=66667,a,b,c,d,e;
4      scanf("%d",&x);
5      while(n>=11111){
6          n--;
7          e=n%10;
8          d=(n/10)%10;
9          c=(n/100)%10;
10         b=(n/1000)%10;
11         a=(n/10000)%10;
12         if((a+b+c+d+e==x)&&(a>0)&&(a<7)&&(b>0)&&(b<7)&&(c>0)&&(c<7)&&(d>0)&&(d<7)&&
(e>0)&&(e<7))
13             printf("%d\n",10000*a+1000*b+100*c+10*d+e);
14     }
15     return 0;
16 }
17 }
```

C.整除

题目描述

yuki舒给了你4个整数 A, B, C, D ($1 \leq A \leq B \leq 10^{18}, 1 \leq C, D \leq 10^9$), 请你求出 $[A, B]$ 中既不能被 C 整除也不能被 D 整除的整数的个数。

输入

多组输入数据

每组数据一行, 分别为4个整数 A, B, C, D

输出

每组数据输出一行, 符合题意的整数的个数

输入样例

```
1 | 4 9 2 3
```

输出样例

```
1 | 2
```

样例解释

5和7符合题意

解题思路

本题 a, b, c, d 的范围都很大。一般认为程序1秒钟的时间内能进行 $10^7 \sim 10^8$ 次运算, 因此本题采用枚举法一定会超时。考虑 $[1, A]$ 内 C 的倍数的个数可以直接由 A/C 求。根据容斥原理, $[1, A]$ 内 c 或 d 的整数倍的个数 $= A/C + A/D - A/\text{lcm}(C, D)$, 这里 lcm 表示最小公倍数。最终答案用是 $[1, B]$ 内的答案数减去 $[1, A-1]$ 内的答案数的差。

AC代码

```
1 | #include <stdio.h>
2 | typedef long long ll;
3 | ll max(ll a, ll b){
4 |     if (a>b){
5 |         return a;
6 |     }
7 |     return b;
8 | }
9 |
10 | ll gcd_r(ll a, ll b){
11 |     int t;
12 |     while(b!=0){
13 |         t=a%b;
14 |         a=b;
15 |         b=t;
16 |     }
17 |     return a;
```

```
18 }
19
20 ll gcd(ll c, ll d){
21     return c*d/gcd_r(c,d);
22 }
23 int main(){
24     long long a,b,c,d;
25     long long count=0;
26     while(scanf("%lld %lld %lld %lld",&a,&b,&c,&d)!=EOF){
27         count=b-a+1- ((b/c-(a-1)/c)+(b/d-(a-1)/d)-(b/gcd(c,d)-(a-1)/gcd(c,d)));
28         printf("%lld\n",count);
29     }
30     return 0;
31 }
```

D.MYH要睡觉

题目描述

MYH学姐最近一直在努力为准备期中考试而努力学习，但她知道劳逸结合才能有更好的效果，所以他决定要睡一会儿觉。

他记录了包括自己在内的五名同学今天的学习时间（单位：s），他要找到几名同学学习时间的中位数，看一看自己学习时间跟大家学习时长的中位数比如何，如果他学的时间比较长，他就要奖励自己多睡一会儿，睡 $8h + (\text{自己的学习时间} - \text{中位数学习时间})s$ ，如果他学的时间比较短，他就要惩罚自己只睡自己学习时间的一半（向下取整），如果他的学习时间恰巧就是中位数，那么他就睡自己学习时间那么多。

输入

共两行。

第一行，一个合法的正整数，表示MYH学姐的学习时长。

第二行，四个合法的正整数，表示其他四名同学的学习时长。

保证这五个整数互不相同。

输出

一个整数，表示MYH学姐可以睡觉的时长（单位：s）。

输入样例1

```
1 | 36000
2 | 36010 36020 36030 36040
```

输出样例1

```
1 | 18000
```

输入样例2

```
1 | 36000
2 | 35010 35020 36010 36020
```

输出样例2

```
1 | 36000
```

输入样例3

```
1 | 36010
2 | 35010 35020 36000 38000
```

输出样例3

```
1 | 28810
```

hint

五个数的中位数是第三大的数。 本题有不用循环结构的快速写法，请探索。

解题思路

本题的一种解法是通过排序找出中位数；这里提供另一种可行的解法，即通过寻找恰好满足 $(a>b), (a>c), (a>d), (a>e)$ 这四个中的2个的数，就可以得到所要找的中位数。

AC代码

```
1  #include<stdio.h>
2  int main(){
3      int a,b,c,d,e,z;
4      scanf("%d%d%d%d%d",&a,&b,&c,&d,&e);
5      if((a>b)+(a>c)+(a>d)+(a>e)==2)z=a;
6      else if((b>a)+(b>c)+(b>d)+(b>e)==2)z=b;
7      else if((c>a)+(c>b)+(c>d)+(c>e)==2)z=c;
8      else if((d>a)+(d>b)+(d>c)+(d>e)==2)z=d;
9      else z=e;
10     if(a==z)printf("%d",a);
11     else if(a<z)printf("%d",a/2);
12     else printf("%d",28800+a-z);
13     return 0;
14 }
```


E. 原码计算

题目描述

给出一个 int 范围内的整数 n ，如果 n 的原码无法表示，则输出 n 的32位补码，否则输出 n 的32位原码。

输入

多组数据输入，每组数据一行，为一个 int 范围内的整数 n 。

输出

对于每组数据，输出一行，一个01串，表示 n 的原码或补码。

输入样例

```
1 | 0
2 | 1
3 | -1
4 | -2147483648
5 | 2147483647
```

输出样例

```
1 | 00000000000000000000000000000000
2 | 00000000000000000000000000000001
3 | 10000000000000000000000000000001
4 | 10000000000000000000000000000000
5 | 01111111111111111111111111111111
```

Hint

1. int 范围内原码无法表示的整数好像只有一个。
2. 计算机存储数据是按补码存储的。
3. 如果 $a = -b$ ，那么 a 和 b 的原码有什么关系呢？仔细想一下怎么把负数用原码表示。

解题思路

在计算机中，数据以补码存储。根据定义，大于等于0时原码等于补码，小于零时，原码数值位等于补码减1取反（ -2^{31} 除外）。实际上我们可以认为在 int 数据类型中，补码是实际数值在 $\text{mod } 2^{32}$ 之后的32位无符号2进制数结果，相加时与无符号32位二进制数相加运算过程相同。同时我们也可以发现，当数值从-1向下减小时，补码的二进制表示由 $111\dots1$ 递减，原码由 $100\dots01$ 递增。这里使用 $\backslash(100\dots0\backslash)_2 + (0 - n)$ 得到原码

对于输出数字的二进制，我们可以利用位运算技巧完成取位操作。 $(n \gg i) \& 1$ 即为将 n 右移 i 位后仅保留最低位，即 n 的第 $i+1$ 位。

AC代码

```
1 #include<stdio.h>
2 int main(){
3     int n, i;
4     while(scanf("%d", &n) != EOF){
5         if(n != 1 << 31 && n < 0) n = (1 << 31) - n; //1000...00 + (-n)
6         for(i = 31; i >= 0; i--){
7             if((n >> i) & 1) putchar('1');
8             else putchar('0');
9         }
10        putchar('\n');
11    }
12 }
```

F. 第一和第二

题目描述

有个名人曾说过，“第二名和最后一名没什么区别，人们只记得第一名”。就像你很容易就能想起世界第一高峰是珠穆朗玛峰，却不容易想起来第二高峰是谁。

输入

一行多个整数，每个整数之前使用空格隔开。

输出

一行两个整数，分别是输入的所有数的最大值和次大值（就是第二大的那个数，即最大值 \neq 次大值），两数之间使用制表符 `\t` 隔开

输入样例

```
1 | 1 50 3 28 45634 12 -34 213 342 4532 546 123435
```

输出样例

```
1 | 123435      45634
```

数据范围

输入的整数在int范围内，可能包含重复的整数

保证最大值和次大值存在

最多输入1000000个整数

Hint

第一高峰：珠穆朗玛峰 ($8844m$)

第二高峰：乔戈里峰 ($8611m$)

第三高峰：干城章嘉峰 ($8586m$)

第四高峰：洛子峰 ($8516m$)

第五高峰：马卡鲁峰 ($8463m$)

解题思路

对于寻找最大值我们可以直接遍历一遍，每个数和当前最大值比较即可。找到第一和第二大的值，则需要在遍历时对新增的数与当前第一、第二大的数比较并维护第一、第二大关系。需要注意的是，这里的第二大值与最大值不可以相等，边界情况需考虑。

这里使用 $1 < 31$ 预置max值为 -2^{-31} ，以保证初始状态下max值不会影响结果，可以结合补码和逻辑左移运算的知识理解

AC代码

```
1  #include<stdio.h>
2  int main(){
3      int n, max1 = 1<<31, max2 = 1<<31;
4      while(scanf("%d", &n) != EOF){
5          if(n > max1){
6              //最大值更新
7              max2 = max1;
8              max1 = n;
9          }
10         else if(n < max1 && n > max2) max2 = n; //次大值更新
11         //n与最大、次大相等情况忽略
12     }
13     printf("%d\t%d", max1, max2);
14 }
15
```

G. 毛毛虫和麻雀

题目描述

小D的花园里养了 n 条毛毛虫，每条毛毛虫都有自己独有的编号（从1到 n ）。

一开始，所有的毛毛虫都开开心心地生活在花园里，享受着食物和阳光。

直到某一天，小D另外一个邻居家养的麻雀发现了这些毛毛虫，于是从那天开始的若干天，这只麻雀每天都会飞过来，吃掉若干只毛毛虫。这只麻雀每天都有一个特别喜欢的数字 x ，而它在那一天**只会吃长度为 x 的毛毛虫**，具体的吃法是这样的：

- 第一天只吃一条毛毛虫；
- 如果前一天吃到了毛毛虫，那么今天只吃一条毛毛虫；
- 如果前一天没有吃到毛毛虫，那么（由于饥饿）今天会把长度为 x 的毛毛虫全部吃掉。

小D想知道，从麻雀发现毛毛虫那一天开始的若干天，每天麻雀吃掉毛毛虫的总长度是多少。

输入

第一行一个正整数 n ，表示一开始有 n 条毛毛虫；

接下来一行有 n 个正整数，其中第 i 个正整数代表了编号为 i 的毛毛虫的长度 l_i （单位：mm）；

接下来一行一个正整数 m ，表示麻雀总共飞来了 m 天；

再接下来一行 m 个正整数，其中第 i 个正整数代表麻雀在第 i 天喜欢的数字。

输入数据保证 $1 \leq n \leq 3e5$ ， $1 \leq l_i, x \leq 1e5$ ， $1 \leq m \leq 5000$ 。

输出

输出 m 行，其中第 i 行代表第 i 天麻雀吃掉的毛毛虫的总长度（单位：mm）。

注意，对于每一条毛毛虫只能被吃一次。

输入样例

```
1 | 4
2 | 1 2 3 3
3 | 3
4 | 2 2 3
```

输出样例

```
1 | 2
2 | 0
3 | 6
```

样例解释

一开始有4条毛毛虫，1号毛毛虫长度为1，2号长度为2，3号长度为3，4号长度为3。麻雀总共飞来了3天。第一天它喜欢的数字是2，所以吃了1条长度为2的毛毛虫，总长度为2；第二天它喜欢的数字还是2，但是已经没有长度为2的毛毛虫了，所以总长度为0；第三天它喜欢的数字是3，由于前一天没吃到毛毛虫，所以它把长度为3的毛毛虫全吃了，总长度为6。

解题思路

注意审题，根据题目描述进行模拟即可。

具体模拟步骤：

1. 记录长度为 x 的毛毛虫个数 $num[x]$;
2. 记录前一天吃的毛毛虫个数 $last$;
3. 在本天吃毛毛虫时，判断能吃多少;
4. 对剩余毛毛虫数量进行调整。

注意分析数据范围，答案可能超过 int 表示的范围，故用 $long\ long$ 输出。

关于逗号运算符，在这里其实是依次执行几个语句，具体请查阅相关资料。

AC代码

```
1  #include<stdio.h>
2  int num[100010];
3  int main(){
4      int i,x,n,m;
5      scanf("%d",&n);
6      for(i=0;i<n;i++){
7          scanf("%d",&x);
8          num[x]++;
9      }
10     scanf("%d",&m);
11     int last=0;
12     for(i=0;i<m;i++){
13         scanf("%d",&x);
14         if(!i){//第一天
15             if(num[x])num[x]--,last=1;
16             else last=0;
17         }else{
18             if(!last)last=num[x],num[x]=0;//前一天没吃到虫
19             else{
20                 if(num[x])last=1,num[x]--;//这一天能吃到虫
21                 else last=0;
22             }
23         }
24         printf("%lld\n",1LL*last*x);
25     }
26     return 0;
27 }
28
```

H.lingxin哥哥的对称差

题目描述

lingxin 鸽鸽最近了解到一个概念，叫对称差。集合 A 和集合 B 的对称差是一个集合，这个集合里的每一个元素要么在 A 中且不在 B 中，要么在 B 中且不在 A 中。例如 $\{1, 2, 3\}$ 和 $\{2, 3, 4\}$ 的对称差是 $\{1, 4\}$ 。再例如 $\{1, 2\}$ 和 $\{1, 2\}$ 的对称差是空集。为了代码实现方便，我们用 $\{\}$ 表示空集。空集所包含的数的个数是 0。

lx 当然对这个概念秒懂，于是他想考考你，给你两个集合，你能输出这两个集合的对称差中所含数的个数吗？

输入

三行。

第一行，由一个空格隔开的两个非负整数 m 和 n ，分别代表集合 A 和集合 B 所包含的数的个数。

第二行和第三行分别代表集合 A 和集合 B 。对于每一行，保证其以前大括号字符 $\{$ 开头、以后大括号字符 $\}$ 结尾，且前后大括号之间有对应个数个非负整数。若集合中数的个数 ≥ 2 ，则每相邻两个数之间必由一个英文逗号隔开。（详细格式可参考样例）

保证每个集合的内部不会有重复的数字。

输出

一行，一个整数，表示输入的两个集合的对称差中所含数字的个数。

输入样例1

```
1 | 3 3
2 | {1,2,3}
3 | {2,3,4}
```

输出样例1

```
1 | 2
```

输入样例2

```
1 | 2 2
2 | {1,2}
3 | {1,2}
```

输出样例2

```
1 | 0
```

输入样例3

```
1 | 0 1
2 | {}
3 | {6}
```

输出样例3

数据范围

保证输入的每个集合所包含的数字个数 $\in [0, 10^6]$

保证输入的每个集合中的每个数均 $\in [0, 10^6]$

Hint

类似第三次上机的士谔数题，数组可以用来对某些值做“访问标记”，来表示这些值曾经出现过。

解题思路

按照提示，使用数组对每个数进行访问标记。一种比较简单的的标记方法是对与a集合中的数，其标记+1，对于b集合中的数，标记+1。最终表示等于1的数的集合就是对称差。

此外本题需要对输入的字符串进行处理，一种方法是用%c过滤掉非数字字符。

本题易错点：

1. 全局变量初值为0，但是定义在函数内的变量初值不定，所以如果标记数组定义在函数内部则需清0。
2. 字符串处理要考虑到换行时可能存在的\r（也可能不存在）。

AC代码

```

1  #include <stdio.h>
2  int s[10000000];
3  int main()
4  {
5      int m,n,i,k,num=0;
6      char x;
7      for(i=0;i<10000000;i++)
8          s[i]=0;
9      scanf("%d %d",&m,&n);
10     scanf("%c\n",&x);
11     scanf("%c",&x);
12     for(i=0;i<m;i++)
13     {
14         scanf("%d",&k);
15         s[k]++;
16         scanf("%c",&x);
17     }
18     if(m==0)
19         scanf("%c",&x);
20     scanf("%c\n",&x);
21     scanf("%c",&x);
22     for(i=0;i<n;i++)
23     {
24         scanf("%d",&k);
25         s[k]++;
26         scanf("%c",&x);
27     }
28     for(i=0;i<10000000;i++)
29         if(s[i]==1)
30             num++;
31     printf("%d",num);
32     return 0;

```


33 | }
34 |

I. MLE

题目描述

可以看到，*OJ*里的每道题目都有内存限制（往往是65536kb）。如果你的程序定义了太多变量，超出内存限制，就会得到新鲜出炉的*MLE*(*Memory Limit Exceed*)。那么，如何计算内存呢——

在C语言里，变量的值以0和1的形式存储在计算机内存中的某个位置。我们的程序不需要知道变量的确切存储位置，因为可以简单地通过变量名引用它们。

程序需要知道每个变量的数据类型。存储简单整数、字母、浮点数的方式是不同的，它们的解释方式不相同，很多时候它们占用的内存也不相同。

C语言中，它们可以分为三种类型：

- 整数类型：它们可以存储整数，如1024或998244353。最基本的是`int`，它占用4个字节。`long long`更大，占用的内存是`int`的两倍。
- 字符类型：它们可以存储单个字符，如M或#。`char`占用1个字节。
- 浮点数类型：它们可以表示实数，如3.1415926535897或1.00，但精度取决于使用哪一种。`float`, `double`占用的内存分别对应`int`和`long long`，而`long double`占用`double`两倍的内存。

现在，你拥有几份C语言代码（编译已经通过了，保证合法）。我们截取了定义变量的语句出来（未提及的任何其他类型保证不出现）。作为一个*freshman*每行只包含一个变量或一个数组，你的任务是计算每一份代码使用的内存总数，以*Kibibyte*为单位输出（1 *Kibibyte*为1024字节）向上取整后的结果。

输入

第1行，一个整数 T ($1 \leq T \leq 100$)表示数据组数。

接下来每组数据的第1行一个整数 n ($1 \leq n \leq 1000$)，表示这份代码的行数。

之后的 n 行，每行一个语句，定义一个变量或数组。形式如`type variable_name;`或`type array_name[array_size];`。`type`会是题面中提到过的数据类型，`name`全部由小写字母组成，是长度不超过10的字符串，`array_size`会是一个不超过 10^5 的正整数。同时我们也保证输入中没有连续的空格。

输出

对于每组数据输出一行Case #x: y, x表示从1开始计数的数据组数，y是一个整数表示以*Kibibyte*为单位的答案（向上取整）。

样例输入

```
1 | 2
2 | 6
3 | char b;
4 | int c;
5 | long long d;
6 | float f;
7 | double g;
8 | long double h;
9 | 1
10 | int a[1000];
```

样例输出

```
1 Case #1: 1
2 Case #2: 4
```

样例解释

第二组数据中，程序占用4000字节内存，转换单位向上取整后答案是4 *Kibibyte*

解题思路

此题主要考察字符串处理。注意到变量类型的首字母互不相同，可以根据首字母判断。

对于数组型变量，可以根据[]来识别，匹配后读取[]后面的数字并与之相乘得到新的数据类型长度。具体方法参考代码

AC代码

```
1 #include<stdio.h>
2 int main(){
3     int t, n, ans, i = 0;
4     scanf("%d", &t);
5     for(; i < t; i++){
6         scanf("%d", &n);
7         ans = 0;
8         char s[10];
9         while(n--){
10             scanf(" %s", s);
11             int k = 0;
12             switch(s[0]){
13                 case 'l':
14                     scanf(" %s", s);
15                     if(s[0] == 'l') k = 8;//long long
16                     else k = 16;//long double
17                     break;
18                 case 'c':k = 1;break;//char
19                 case 'd':k = 8;break;//double
20                 default :k = 4;break;//int float
21             }
22             char c;
23             while((c = getchar())!= ';'){
24                 if(c == '['){
25                     int sz;
26                     scanf("%d", &sz);
27                     k *= sz;
28                 }
29             }
30             ans += k;
31         }
32         printf("Case #d: %d\n", i, (ans + 1023)/ 1024);
33     }
34 }
35
```

J.HugeGun学姐疯了

题目描述

HugeGun学姐疯了。她给了你一个数 x ($1 \leq x \leq 100$)。要你输出一个由 x 行字符构成的图形：

```
1      $
2     $$$
3    $$$$$$
4   $$$$$$@$$$$$
```

这是一个左右对称的图形，其中，第 i 有 i^2 个美元符号，如果 i 为偶数，则在正中间加一个'@'以保证对称。

输入

一行，一个整数 x

输出

对于每组数据，输出 x 行，表示这个图形

输入样例

```
1 | 4
```

输出样例

```
1      $
2     $$$
3    $$$$$$
4   $$$$$$@$$$$$
```

解题思路

本题利用循环嵌套即可完成，比较简单，需要注意奇偶行的输出，不同并在代码中加以体现。

AC代码

```
1 #include <stdio.h>
2 int main(){
3     int x,i,j;
4     scanf("%d",&x);
5     for(i=1;i<=x;i++){
6         if(i%2==1){
7             for(j=1;j<=x*x/2-i*i/2;j++){
8                 printf(" ");
9             }
10            for(j=1;j<=i*i;j++){
11                printf("$");
12                if(j==i*i)
13                    printf("\n");
14            }
15        }else{
16            for(j=1;j<=x*x/2-i*i/2;j++){
17                printf(" ");
```

```
18         }
19         for(j=1;j<=i*i/2;j++){
20             printf("$");
21         }
22         printf("@");
23         for(j=1;j<=i*i/2;j++){
24             printf("$");
25             if(j==i*i/2)
26                 printf("\n");
27         }
28     }
29 }
30 return 0;
31 }
32
```