

# A. 我要AK!

## 解题思路

分支结构 & 转义字符。分别针对  $t = 0$ ,  $t = 1$  和其他情况输出即可。AK 字符画需要将 \ 替换为 \\。

## AC代码

```
1  #include<stdio.h>
2  int main(){
3      int t;
4      scanf("%d", &t);
5      if(t == 0){
6          printf(
7              "  -----  \n"
8              "  *****  \n"
9              "  -----  \n"
10             "****      ***  \n"
11             "-----  \n"
12             "*****      ***  \n"
13             "-----  \n"
14             "****      ***** ");
15     }
16     else if(t == 1){
17         printf(
18             "              ,--. \n"
19             "      ,---,      ,--/  / \n"
20             "      ' . ' \\      ,---,': / ' \n"
21             "      / ;      ' . : : '/ / \n"
22             "      : :      \\ | ' , \n"
23             "      : | /\ \  \\ | ' / \n"
24             "      | : ' ; . : | ; ; \n"
25             "      | | ;/ \\  \\ : ' \\ \n"
26             "      ' : | \\ \\ , ' | ' \n"
27             "      | | ' '---' ' : |. \\ \n"
28             "      | : :      | | ' _\\ \n"
29             "      | | , '      : | \n"
30             "      `---'      ; |, ' \n"
31             "              '---' ");
32     }
33     else{
34         printf("AC is the first step to AK.");
35     }
36 }
```

## B. yuki舒的电源插口

### 解题思路

设接线板个数为 $x$ ，满足题目要求的不等式为 $A + (A - 1) * (x - 1) \geq B$ ，解出 $x \geq \frac{B-A}{A-1} + 1$ ，因为 $x$ 为整数，取 $x = \lceil \frac{B-A}{A-1} \rceil + 1$ （向上取整）即可。

hint：当求 $\lceil \frac{a}{b} \rceil$ 时，利用整数除法向下取整的特性，等价于求 $\lfloor \frac{a+b-1}{b} \rfloor$ ，故本题式子可化为 $x = \lfloor \frac{B+A-3}{A-1} \rfloor$ 。

### AC代码

```
1 #include<stdio.h>
2 int main(){
3     int a,b;
4     scanf("%d%d",&a,&b);
5     printf("%d", (b+a-3)/(a-1));
6     return 0;
7 }
```

## C. 花式除法

### 解题思路

如hint, 在c语言程序中整数除法默认丢掉整数部分, 所以我们只要运用好整数除法进行判断即可。但是一定要注意正负数的向上和向下取整的问题和是否整除的问题。

另: 同样可以使用类似B题中的向上取整方式进行计算。

### AC代码

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

# D. Ganten学姐的复数计算器

## 解题思路

本题主要考察的内容是分支语句，按照题意完成每一个分支即可。注意读入运算符时%c前面要打一个空格，表示跳过空白字符。

hint1: %c前没空格，scanf()将读取标准输入流中的第一个字符，%c前有空格，scanf()则读取标准输入流中第一个非空白字符。

hint2: 也可以用字符串读入的方式读入一个字符串(%s)，然后取字符串的第一位为结果(参看第二次上机E题的hint)。

## AC代码

```
1  #include <stdio.h>
2
3  int main() {
4      double a,b,c,d;
5      char opt;
6      while (scanf("%lf%lf%lf%lf %c",&a,&b,&c,&d,&opt) != EOF) {
7          if (opt == '+') {
8              printf("%.21f",a+c);
9              if (b+d >= 0)
10                 printf("+");
11                 printf("%.21fi\n",b+d);
12             }
13             else if (opt == '-') {
14                 printf("%.21f",a-c);
15                 if (b-d >= 0)
16                     printf("+");
17                     printf("%.21fi\n",b-d);
18             }
19             else if (opt == '*') {
20                 printf("%.21f",a*c-b*d);
21                 if (a*d+b*c >= 0)
22                     printf("+");
23                     printf("%.21fi\n",a*d+b*c);
24             }
25             else if (c == 0 && d == 0)
26                 printf("??+??i\n");
27             else {
28                 printf("%.21f",(a*c+b*d)/(c*c+d*d));
29                 if ((b*c-a*d)/(c*c+d*d) >= 0)
30                     printf("+");
31                 printf("%.21fi\n",(b*c-a*d)/(c*c+d*d));
32             }
33         }
34         return 0;
35     }
```

## E. 闰年判断

### 解题思路

某个年份是闰年当且仅当：

1. 是400的倍数；
2. 是4的倍数，且不是100的倍数。

用条件语句判断或直接使用逻辑表达式即可。（另一种逻辑表达式见PPT）

### AC代码1

```
1  #include<stdio.h>
2  int main(){
3      int n;
4      while(~scanf("%d",&n))
5          printf("%d\n",(! (n%4)&& n%100) || ! (n%400));
6      //这里存在着隐式类型转换，n%4的bool值为表达式n%4!=0的值。
7  }
```

### AC代码2

```
1  #include<stdio.h>
2  int main(){
3      int n;
4      while(~scanf("%d",&n)){
5          if(n%400==0)printf("1\n");
6          else if(n%100==0)printf("0\n");
7          else if(n%4==0)printf("1\n");
8          else printf("0\n");
9      }
10 }
```

## F. xx有点二

### 解题思路

想象一个数为二进制表示是什么样子。比如样例： $17 = (10001)_2$ ，除以2的余数即是其最后一位二进制的值，商为 $8 = (1000)_2$ ，即前几位的二进制表示。

于是用一个`while`循环，每次取出当前数二进制表示的最后一位，并将当前数除以2，当当前数为0时跳出循环。

### AC代码1

```
1  #include<stdio.h>
2  int main(){
3      int x,cnt=0;
4      scanf("%d",&x);
5      while(x){
6          cnt+=x%2;
7          x/=2;
8      }
9      printf("%d",cnt);
10     return 0;
11 }
```

另外介绍一个内置函数：`__builtin_popcount(x)`，求出 $x$ 的二进制表示下有多少个1。

### AC代码2

```
1  #include<stdio.h>
2  int main(){
3      int x;
4      scanf("%d",&x);
5      printf("%d",__builtin_popcount(x));
6      return 0;
7  }
```

# G. 一元二次方程求解

## 解题思路

我们使用求根公式的朴素方法编写求解程序。首先计算出 $\Delta$ ，然后根据计算结果按要求进行输出，小于零则输出无解，等于零则输出两个相同解 $-\frac{b}{2a}$ （程序中： $-b/(2*a)$ ），大于0则输出两个解。

注意在两个不同根时，大小不可以仅仅根据加减根号 $\Delta$ 决定，还受 $a$ 的正负影响。书写表达式时注意运算优先级，乘除高于加减，同优先级时从左向右运算，不可写成 $-b/2*a$ 。

这道题在这种方法下不会受浮点数精度影响而导致评测结果WA。但日常运算中，如果判断中涉及浮点数相等的边界情况，应引入一个极小的浮点数 $eps$ ，略微放宽约束范围，否则由于十进制浮点数转二进制浮点数和一些函数运算产生的误差会在运算过程中被放大，判断结果将受到影响。

## AC代码

```
1  #include<stdio.h>
2  #include<stdio.h>
3  #define eps 1e-10
4  int main(){
5      double a,b,c;
6      scanf("%lf %lf %lf", &a, &b, &c);
7      double d = b*b - 4*a*c;
8      if(d < -eps)
9          printf("No real roots.");
10     else if (d <= eps)
11         printf("x1=x2=%.21f", -b/2/a);
12     else{
13         double u = (-b + sqrt(d))/2/a, v = (-b - sqrt(d))/2/a;
14         if(u < v)printf("x1=%.21f,x2=%.21f", u, v);
15         else printf("x1=%.21f,x2=%.21f", v, u);
16     }
17     return 0;
18 }
```

# H. 素数判断

## 解题思路

考虑采用枚举约数的方法来判断是否是素数，对于一个正整数 $x$ ，如果存在大于1小于 $x$ 的约数，就不是素数，否则是。求一个数的所有约数同样使用枚举法。枚举可以通过循环实现。这里采用了 $for$ 循环，第一个分号前是初始化语句，只会在进入循环前执行一次。第一个分号和第二个分号之间是循环终止条件，在每一次循环开始前执行，若返回值不为0，则终止循环。第二个分号后是在每次循环后执行的语句。这里同样可以使用 $while$ 循环。

hint：为什么枚举到10000就行了呢，很明显如果 $\sqrt{n}$ 内不存在除1以外的 $n$ 的因数，那么 $n$ 就是质数了。观察题目中给的数据范围， $\sqrt{100000000} = 10000$ 。

## AC代码

```
1  #include <stdio.h>
2
3  int main() {
4      int i,a,b,isp_a = 1,isp_b = 1;
5      scanf("%d%d",&a,&b);
6      for (i = 2;i <= 10000; i++) {
7          if (a%i == 0 && i < a)
8              isp_a = 0;
9          if (b%i == 0 && i < b)
10             isp_b = 0;
11     }
12     if (isp_a == 1 && isp_b == 1) {
13         printf("Yes! That is good!\n");
14         for (i = 1;i <= b-a; i++)
15             if ((b-a)%i == 0)
16                 printf("%d ",i);
17     }
18     else {
19         printf("No! This is not suitable!\n");
20         if (isp_a == 0)
21             printf("%d ",a);
22         if (isp_b == 0)
23             printf("%d ",b);
24     }
25     return 0;
26 }
```



# I. 山顶的星星

## 解题思路

针对图形中的对称情况我们可以用循环来控制输出。对于第一行和最后一行特殊情况用单循环控制，剩余规律情况使用双重循环输出即可。

注意空格和换行，否则会PE。

## AC代码

```
1  #include <stdio.h>
2  int main()
3  {
4      int n;
5      int i, j;
6      //第一行
7      scanf("%d", &n);
8      for(i = 0; i < n - 1; i++) printf(" ");
9      printf("*");
10     for(i = 0; i < n - 1; i++) printf(" ");
11     printf("\n");
12     //中间部分
13     for(i = 1; i < n; i++)
14     {
15         for(j = 0; j < n - i; j++) printf(" ");
16         for(j = 0; j < i; j++) printf("& ");
17         for(j = 0; j < n - i - 1; j++) printf(" ");
18         printf("\n");
19     }
20     //末尾
21     for(i = 1; i < n; i++) printf("& ");
22     printf("&");
23 }
```

# J. 多项式求和

## 解题思路

本题可以用数组实现，也可以不用。

不用数组实现可以考虑函数 $\text{pow}(x,y)$ 即 $x$ 的 $y$ 次幂，本函数出现在`#include<math.h>`函数库中。应当注意的是，应当尽量少用`math.h`的库函数，浮点误差容易产生错误，但本题数据范围较小，误差在允许的范围内。

另外要注意数据叠加后会超过`int`需要用`long long`保存结果

使用数组实现可以先记录下各项系数，从后向前递推实现秦九韶算法。

## AC代码1

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      int n, x, i, a;
6      long long sum = 0;
7      scanf("%d%d", &n, &x);
8      for (i = 0; i <= n; i++) {
9          scanf("%d", &a);
10         sum = sum + a * (long long) pow(x, i);
11     }
12     printf("%lld", sum);
13     return 0;
14 }
```

## AC代码2

```
1  #include <stdio.h>
2
3  int main() {
4      int n, x, i, a[100];
5      long long int sum = 0;
6      scanf("%d%d", &n, &x);
7      for (i = 0; i <= n; i++) {
8          scanf("%d", &a[i]);
9      }
10     for (i = n; i >= 0; i--) {
11         sum = sum * x + a[i];
12     }
13     printf("%lld", sum);
14     return 0;
15 }
```

# K. triangle

## 解题思路

本题考察了库函数和浮点数类型的应用。

根据海伦公式  $S = \sqrt{p(p-a)(p-b)(p-c)}$ , 其中  $p = \frac{a+b+c}{2}$ , 可以直接算出答案, 注意本题要求六位精度, 使用 *float* 型精度不足, 要使用 *double* 型。

hint: 一般情况下, 表示浮点数的时候尽量用 *double* 而非 *float*, 以保证精度等问题。

## AC代码

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int a,b,c,n;
7      scanf("%d",&n);
8      while(n-->0)
9      {
10         scanf("%d%d%d",&a,&b,&c);
11         if(a+b<=c || a+c<=b || b+c<=a) printf("Oh, no!\n");
12         else
13         {
14             double p=1.0*(a+b+c)/2;
15             printf("%.6f\n",sqrt(p*(p-a)*(p-b)*(p-c)));
16         }
17     }
18
19     return 0;
20 }
```