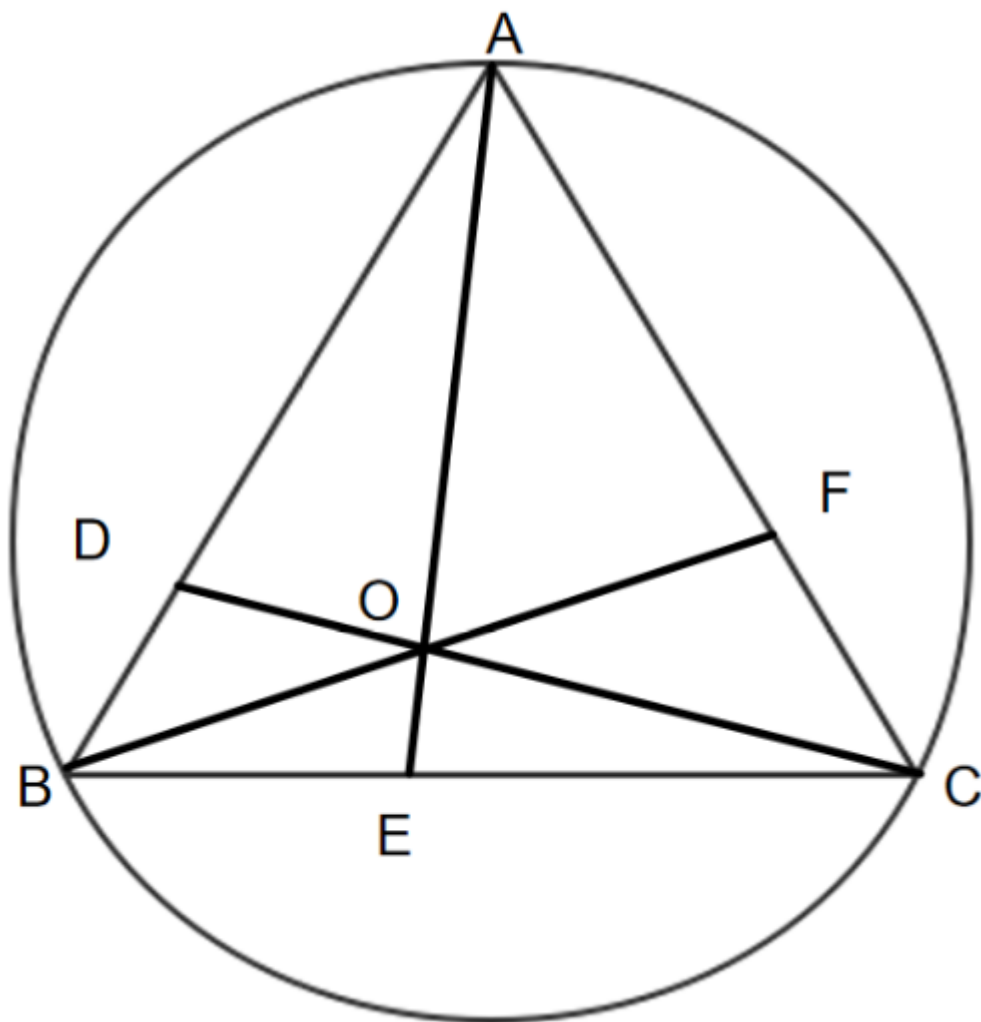


A、画个圈圈诅咒你

这道题就是一个“简单”的数学题，先把图祭出来。



首先，我们先来看下这个圆，整道题看来这题貌似跟圆没有关系，但是会不会有什么暗示？答案是肯定不存在的。

显而易见，题目给出的公式即是塞瓦定理，通过正弦定理转换一下得出塞瓦定理的角元形式：

塞瓦定理适用于任意三角形，证明自行百度。

$$\frac{\sin \angle BAO}{\sin \angle OAC} \cdot \frac{\sin \angle ACO}{\sin \angle OCB} \cdot \frac{\sin \angle CBO}{\sin \angle OBA} = 1$$

化下简

$$\frac{\sin \angle BAO}{\sin(\frac{\pi}{3} - \angle BAO)} = \frac{\sin(\frac{\pi}{3} - \angle ACO)}{\sin \angle ACO} \cdot \frac{\sin(\frac{\pi}{3} - \angle CBO)}{\sin \angle CBO}$$

为了简化计算，我们将三个角重命名 a 、 b 、 c

$$\frac{\sin a}{\sin(\frac{\pi}{3} - a)} = \frac{\sin(\frac{\pi}{3} - b)}{b} \cdot \frac{\sin(\frac{\pi}{3} - c)}{\sin c}$$

等式右边的值是已知的，为方便计算，令等式右边值等于 tmp 。注意输入为角度，运算时转换为弧度计算，假设输入 $\angle BAO$ 度数为 60° 则相应转换公式为 $\sin \angle BAO = \sin(\frac{60}{180} \cdot \pi)$ 。防止精度丢失可令 $\pi = \text{acos}(-1)$ 。

再次化简—(化简过程略，太简单子)—

$$T = \sin a = \sqrt{\frac{\frac{3}{4} \cdot tmp^2}{tmp^2 + tmp + 1.0}}$$

然后反三角函数答案就出来了

$$\angle BAO = a = \text{asin} T$$

代码：

```
1  #include<stdio.h>
2  #include<math.h>
3  #define PI acos(-1)
4  int main() {
5      int t;
6      double a, b, tmp, x, ans;
7      scanf("%d", &t);
8      while(t--) {
9          scanf("%lf%lf", &a, &b);
10         tmp = sin(PI / 3.0 - a / 180.0 * PI) / sin(a / 180.0 *
PI) * sin(PI / 3.0 - b / 180.0 * PI) / sin(b / 180.0 * PI);
11         x = sqrt(3.0 / 4.0 * tmp * tmp / (tmp * tmp + tmp +
1.0));
12         ans = asin(x) / PI * 180.0;
13         printf("%.2f\n", ans);
```

```
14     }
15     return 0;
16 }
```

B、咸鱼落泪

本题是一个排序问题，由于数据范围很小，可以用各种姿势过。大家有时间可以去学习下各种排序，本题题解采用一个最简单易懂的排序方法--冒泡排序。

代码：

```
1  #include<stdio.h>
2  int main() {
3      int a[105] = {0}; //一般比题目数据范围要求大一点
4      int i, j, t, n;
5      //数据输入
6      scanf("%d", &n);
7      for(i = 0; i < n; i++) {
8          scanf("%d", a + i);
9      }
10     //冒泡排序
11     for(i = 0; i < n - 1; i++) { //n个数的数列总共扫描n-1次
12         for(j = 0; j < n - i - 1; j++) { //每一趟扫描到a[n-i-2]与
13             //a[n-i-1]比较为止结束
14             if(a[j] > a[j + 1]) { //后一位数比前一位数小的话，就交换
15                 //两个数的位置（升序）
16                 t = a[j + 1];
17                 a[j + 1] = a[j];
18                 a[j] = t;
19             }
20         }
21     }
22     //输出排列
23     for(i = 0; i < n; i++) {
24         printf("%d ", a[i]);
25     }
26     return 0;
27 }
```

C、World

这题也没什么好说的，就是考察一下 `if` 判断和输入输出格式控制，先计算出总价钱，然后再判断是否有优惠，如果有优惠减去优惠金额即可。

代码：

```
1  #include<stdio.h>
2  int main() {
3      double n, sum = 0;
4      scanf("%lf", &n);
5      sum = n * 6;
6      if(n >= 15)
7          sum -= n * 0.1;
8      printf("%.2f", sum);
9      return 0;
10 }
```

D、Hello

这题你没看错，是送分题，如果你挨着挨着把题目读完了，你会发现已经有人过这道题。题目要求输出一行 `Hello, WITACM!` 注意写代码直接复制粘贴。

代码：

```
1  #include<stdio.h>
2  int main() {
3      printf("Hello, WITACM!\n");
4      return 0;
5  }
```

E、白色相簿

题意：

- 判断是否存在3人以上的多元关系

解题思路：

- 因为不能存在3人以上的关系，即每人最多只能和除自己以外的1个人存在关系

所以记录一下每个人和几个人存在关系，只要存在有人关系数 ≥ 2 即不稳定

- 先将题目给出的所有关系排序，删除重复出现的关系，和自身与自身之间的关系。将剩下的关系进行计数，每存在一对关系，两个人的计数+1。
- 最后对所有人进行遍历，看是否存在大于等于2的人，如果存在，说明这个人至少与2个不同的人之间存在关系，输出Error；否则输出Nice
- 例如：A和B存在关系，A和B计数+1；A和C存在关系，A和C计数+1。最后A计数为2，存在两个不同关系，即不稳定，输出Error

代码：

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  struct abc
5  {
6      int x,y,f;
7  }s[10005];
8
9  int a[10005];
10
11 bool cmp(abc a,abc b)
12 {
13     return (a.x<b.x || (a.x==b.x && a.y<b.y));
14 }
15
16 int main()
17 {
18     int i,j,m,n,t,x,y,xx,yy;
```

```

19     cin>>n>>m;
20     for (i=1;i<=m;i++)
21     {
22         cin>>x>>y;
23         xx=min(x,y);
24         yy=max(x,y);
25         s[i].x=xx;
26         s[i].y=yy;
27         s[i].f=1;
28     }
29     sort(s+1,s+m+1,cmp);
30     for (i=1;i<m;i++)
31     {
32         if (s[i].x==s[i+1].x && s[i].y==s[i+1].y)
33             s[i+1].f=0;
34     }
35     memset(a,0,sizeof(a));
36     for (i=1;i<=m;i++)
37         if (s[i].f && s[i].x!=s[i].y)
38         {
39             a[s[i].x]++;
40             a[s[i].y]++;
41         }
42     bool ans=true;
43     for (i=1;i<=n;i++)
44         if (a[i]>=2)
45         {
46             ans=false;
47             break;
48         }
49     if (ans)
50         cout<<"Nice"<<endl;
51     else
52         cout<<"Error"<<endl;
53     return 0;
54 }

```

F、英雄联盟

题意：

- 题目描述较长，按照描述规则进行八强抽签模拟，注意理解同组规避原则

解题思路：

- 按照规则直接模拟即可，先放位置1，从小组第一中选1个；然后位置2，从小组第二中选一个；位置3从剩下3个小组第一中选1个；位置4从剩下3个小组第二中选1个；位置5从剩下2个小组第一中选1个……以此类推
- 选取时优先按照字典序顺序选取（从小到大），最终结果共有96种。

代码：

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  string s1[4]={"AFS","FNC","KT","RNG"};
5  string s2[4]={"C9","EDG","G2","IG"};
6  string s[1000];
7  map<string,int> a;
8
9  int main()
10 {
11     a["AFS"]=1; a["G2"]=1;
12     a["RNG"]=2; a["C9"]=2;
13     a["KT"]=3; a["EDG"]=3;
14     a["FNC"]=4; a["IG"]=4;
15     int i,i1,i2,i3,i4,j1,j2,j3,j4,x=0;
16     string t1,t2,t3,t4,t5,t6,t7,t8,tt;
17     for (i1=0;i1<4;i1++)
18     {
19         t1=s1[i1];
20         for (j1=0;j1<4;j1++)
21         {
22             t2=s2[j1];
23             for (i2=0;i2<4;i2++)
24                 if (i2!=i1)
25                 {
26                     t3=s1[i2];
27                     for (j2=0;j2<4;j2++)
28                         if (j2!=j1)
```

```

29         {
30             t4=s2[j2];
31             for (i3=0;i3<4;i3++)
32                 if (i3!=i1 && i3!=i2)
33                     {
34                         t5=s1[i3];
35                         for (j3=0;j3<4;j3++)
36                             if(j3!=j1 && j3!=j2)
37                                 {
38                                     t6=s2[j3];
39                                     for (i4=0;i4<4;i4++)
40                                         if ((i4!=i3) &&
41                                             (i4!=i2) && (i4!=i1))
42                                                 {
43                                                     t7=s1[i4];
44                                                     for
45                                                         if
46                                                             {
47                                                                 if (
48                                                                     (a[t1]!=a[t2] && a[t1]!=a[t3] && a[t1]!=a[t4] && a[t2]!=a[t3] &&
49                                                                     a[t2]!=a[t4] && a[t3]!=a[t4]) && (a[t5]!=a[t6] && a[t5]!=a[t7] &&
50                                                                     a[t5]!=a[t8] && a[t6]!=a[t7] && a[t6]!=a[t8] && a[t7]!=a[t8]) )
51                                                                     {
52                                                                 tt=t1+" "+t2+" "+t3+" "+t4+" "+t5+" "+t6+" "+t7+" "+t8;
53                                                                 x++;
54                                                                 s[x]=tt;
55                                                                 }
56                                                             }
57                                                         }
58                                 }
59                         }
60                     }

```



```

61     cout<<x<<endl;
62     for (i=1;i<=x;i++)
63         cout<<s[i]<<endl;
64     return 0;
65 }
66

```

G、禁言大冒险

题意：

- 禁言规则如题目描述

解题思路：

此题需要分情况讨论，首先是两种大情况：

- 一、打断复读的和复读最后一人是同一人，禁言最后一个人
- 二、打断复读的和复读最后一人不同，禁言倒数第二个人

对于这两种大情况，每种又细分为4种小情况：

- 1、群主（yellow）在，应该禁言的人为群主时，此时向后顺延一个（不管下一个是管理员还是普通群员都直接禁言）；
- 2、群主不在，应该禁言的为管理员（green）时，此时向后顺延一直找到下一个普通群员
- 3、群主在，应该禁言的为管理员或普通群员时，直接禁言
- 4、群主不在，应该禁言的为普通群员时，直接禁言

在读入数据时先记录群主是否参与复读，然后再分具体情况输出禁言对象

代码：

```

1  #include<bits/stdc++.h>

```

```
2 using namespace std;
3
4 int main()
5 {
6     int i,j,m,n,t,s;
7     string s1[105],s2[105];
8     string x,y;
9     cin>>n;
10    t=0;
11    for (i=1;i<=n;i++)
12    {
13        cin>>s1[i]>>s2[i];
14        if (s2[i][0]=='y')
15            t=1;
16    }
17    cin>>x>>y;
18    if (x==s1[n])
19    {
20        if (y[0]=='y')
21            cout<<s1[n-1]<<endl;
22        else if (y[2]=='e' && t!=1)
23        {
24            for (i=n-1;i>=1;i--)
25            {
26                if (s2[i][2]=='a')
27                {
28                    cout<<s1[i]<<endl;
29                    break;
30                }
31            }
32        }
33        else
34            cout<<x<<endl;
35    }
36    else
37    {
38        if (s2[n-1][0]=='y')
39            cout<<s1[n-2]<<endl;
40        else if (s2[n-1][2]=='e' && t!=1)
41        {
42            for (i=n-2;i>=1;i--)
43            {
```

```

44         if (s2[i][2]=='a')
45         {
46             cout<<s1[i]<<endl;
47             break;
48         }
49     }
50 }
51 else
52     cout<<s1[n-1]<<endl;
53 }
54 return 0;
55 }

```

H、刺客信条

题意：

- 从S点出发，到E点，中间经过每个点需要花费时间，经过A、B、C三点时花费时间为100，求最短用时

解题思路：

- 该题为一道典型的搜索题，搜索从起点到终点的最短路径。
- 可以在读入处理时直接将A、B、C处理为100，然后进行广度优先搜索
- 在搜索过程中每走到终点一次，将总用时记录下来，如果小于之前总用时，则更新。凡是大于当前最小总用时的其他道路可以直接放弃（因为继续走下去不可能总用时小于当前用时了）
- 直到将所有路走完一遍，或剩下的路全都超过当前总用时，搜索终止，输出最终结果。

代码：

```

1  #include<bits/stdc++.h>
2  using namespace std;
3

```

```

4  int ans,t,n,m;
5  int sx,sy,ex,ey;
6  int a[105][105]={0};
7  int b[105][105]={0};
8  int f[4][2]={{1,0},{-1,0},{0,1},{0,-1}};
9
10 struct abc
11 {
12     int x,y,t;
13 };
14
15 void BFS(int x,int y)
16 {
17     queue<abc> q;
18     abc start;
19     start.x=x,start.y=y,start.t=0;
20     q.push(start);
21     while (!q.empty())
22     {
23         abc front=q.front();
24         q.pop();
25         if (front.x<1 || front.x>n || front.y<1 || front.y>m)
26             continue;
27         else if (front.x==ex && front.y==ey)
28         {
29             if (ans>front.t)
30                 ans=front.t;
31             continue;
32         }
33         else if (front.t>ans)
34         {
35             b[front.x][front.y]=1;
36             continue;
37         }
38         else
39         {
40             b[front.x][front.y]=1;
41             int x1,y1;
42             for (int i=0;i<4;i++)
43             {
44                 x1=front.x+f[i][0];
45                 y1=front.y+f[i][1];

```

```

46         if (!b[x1][y1])
47         {
48             abc v;
49             v.x=x1;
50             v.y=y1;
51             v.t=front.t+a[x1][y1];
52             q.push(v);
53         }
54     }
55 }
56 }
57 }
58
59 int main()
60 {
61     int i,j;
62     char c;
63     cin>>n>>m;
64     for (i=1;i<=n;i++)
65         for (j=1;j<=m;j++)
66         {
67             cin>>c;
68             if (c=='S')
69                 sx=i,sy=j;
70             else if (c=='E')
71                 ex=i,ey=j;
72             else if (c=='A' || c=='B' || c=='C')
73                 a[i][j]=100;
74             else
75                 a[i][j]=c-'0';
76             getchar();
77         }
78     ans=10000,t=0;
79     BFS(sx,sy);
80     cout<<ans<<endl;
81     return 0;
82 }

```

I、太吾绘卷

题意：

- 按照题目描述计算伤害，注意是回合制游戏

解题思路：

- 先求出内力总量和U，然后根据公式计算出S与T，再计算出每回合造成的伤害W1和受到的伤害W2
- 判断一下能否破防，不能则直接输出No
- 如果可以破防，你打死敌人需要的回合数 = 敌人血量V2/你的伤害W1，敌人打死你的回合数 = 你的血量V1/敌人伤害W2，判断一下你能否在被打死之前打死敌人

注意最后输出结果时，如果你用了n回合打死敌人，那么我只受到了敌人n-1回合的伤害

代码：

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  int main()
5  {
6      int i,j,m,n,u1=0,u2=0;
7      double s1,s2,t1,t2,w1,w2,v1,v2;
8      cin>>n;
9      for (i=1;i<=n;i++)
10     {
11         cin>>m;
12         u1+=(10-m)*10;
13     }
14     cin>>u2>>v1>>v2;
15     s1=u1*1.0/500;
16     t1=u1*1.0/1000;
17     s2=u2*1.0/500;
18     t2=u2*1.0/1000;
```

```

19     if (1+s1<=t2 && 1+s2<=t1)
20     {
21         cout<<"No"<<endl;
22         return 0;
23     }
24     else if (1+s1<=t2 && 1+s2>t1)
25     {
26         cout<<"No"<<endl;
27         return 0;
28     }
29     else if (1+s2<=t1 && 1+s1>t2)
30     {
31         printf("%.2lf\n",v1);
32         return 0;
33     }
34     else
35     {
36         w1=u1*(1+s1-t2);
37         w2=u2*(1+s2-t1);
38         m=ceil(v2/w1);
39         v1=v1-(m-1)*w2;
40         if (v1>0)
41             printf("%.2lf\n",v1);
42         else
43             cout<<"No"<<endl;
44     }
45     return 0;
46 }
47

```

J、玩UNO还是玩斗地主

题意：

- 给出两人出石头、剪刀、布的概率，计算两人获胜的概率

解题思路：

- 小A获胜概率 = 小A石头小B剪刀 + 小A剪刀小B布 + 小A布小B石头
- 即 $ans_1 = p_1 * q_2 + p_2 * q_3 + p_3 * q_1$
- 小B获胜概率 = 小B石头小A剪刀 + 小B剪刀小A布 + 小B布小A石头
- 即 $ans_2 = q_1 * p_2 + q_2 * p_3 + q_3 * p_1$
- $ans_1 > ans_2$ 时小A获胜概率更大, $ans_1 < ans_2$ 时小B获胜概率更大,
 $ans_1 == ans_2$ 时概率相等

代码:

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  int main()
5  {
6      double p1,p2,p3,q1,q2,q3;
7      double ans1,ans2,ans3;
8      while(~scanf("%lf%% %lf%% %lf%%",&p1,&p2,&p3))
9      {
10         p1/=100; p2/=100; p3/=100;
11         scanf("%lf%% %lf%% %lf%%",&q1,&q2,&q3);
12         q1/=100; q2/=100; q3/=100;
13         ans1=p1*q2+p2*q3+p3*q1;
14         ans2=p1*q1+p2*q2+p3*q3;
15         ans3=p2*q1+p3*q2+p1*q3;
16         if (ans1>ans3)
17             cout<<"A"<<endl;
18         else if (ans1<ans3)
19             cout<<"B"<<endl;
20         else
21             cout<<"equal"<<endl;
22     }
23     return 0;
24 }
```