算法集目录

/*旅行家的预算问题

描述 Description

一个旅行家想驾驶汽车以最少的费用从一个城市到另一个城市(假设出发时油箱是空的)。给定两个城市之间的距离 d1、

汽车油箱的容量 c（以升为单位)、每升汽油能行驶的距离 d2、出发点每升汽油价格 p 和沿途油站数 n，油站 i 离出发点

的距离 d[i]、每升汽油价格 p[i]。

输入格式 Input Format

输入共 n+1 行，第一行为 d1,c,d2,p,n，以下 n 行，每行两个数据，分别表示该油站距出发点的距离 d[i]和该油站每升

汽油的价格 p[i]。两个数据之间用一个空格隔开。

输出格式 Output Format

一行，输出最少费用。

计算结果四舍五入至小数点后两位。

如果无法到达目的地，则输出-1。

样例输入 Sample Input

```
275.6 11.9 27.4 2.8 2
102.0 2.9
220.0 2.2
```

样例输出 Sample Output

```
26.95
```

【分析】

贪心，记住这种思想。遇见一到题，特别求最优解的题，要考虑它是否可以通每步选择最优来达到最后的全局最优，

对于本题，想想，跑到贵的地方加油就不划算，那现在就一最小的代价，跑到最便宜的地方去，所以刚好加够油就

行了，如果当前站已经很便宜了，那当然不能错过这种好机会了，加满它，跑到最远的相对的最便宜的地方，在重

复这种抉择过程，实际经验告诉我们，这样到最后一定是全局最优的

```
*/
#include<stdio.h>
#include<iostream>
using namespace std;
double p[101],dist[101];
double c,dic;
int N;
double cost,rest,need;
void init()
{
    double box,dd,pp;
    int k;

    scanf("%lf%lf%lf%lf%d",&box,&c,&dic,&p[0],&N);
    dist[N+1]=box;p[N+1]=0;dist[0]=0;
    for(int i=1;i<=N;i++)
    {
        scanf("%lf%lf",&dd,&pp);
        p[i]=pp;dist[i]=dd;
    }
}
void wo()
{

    int j,k,min1,min2;
    rest=cost=k=0;
    while(k<=N)
    {
        j=k;min1=0;min2=0;

        while((dist[j+1]-dist[k]<=c*dic)&&(j<=N))
        {
            j++;
            if(min1==0    &&    p[j]<p[k]) min1=j;
            if(min2==0    ||    p[j]<p[min2]) min2=j;
        }
        if(j==k)
        {
            printf("No solution\n");
            exit(0);
        }
        if(min1!=0)
        {

            need=(dist[min1]-dist[k])/dic-rest;
            if(need<0) need=0;
            cost+=need*p[k];
            rest=0;
            k=min1;
        }
        else
        {
            need=c-rest;
            cost+=need*p[k];
            rest=c-(dist[min2]-dist[k])/dic;
            k=min2;
        }
    }
    return;
}
int main()
{
    init();
    wo();
    printf("%0.2lf\n",cost);
    return 0;
}
```

****

****

登山机器人问题

```
#include<stdio.h>
//贪心法
int b[6][11];
int n,k,m;
int x[6];
int fun()
{
   int i,min=100000000,j;
   for(i=1; i<=n; i++)
     {
        if(x[i]<=k)
          {
            if(min > b[i][x[i]]-b[i][x[i]-1])
              {
                min=b[i][x[i]]-b[i][x[i]-1];
                j=i;
              }
          }
     }
   x[j]++;
   return min;
}

int main()
{
    int i,j,l,c1,c2,sum;

while(scanf("%d%d%d",&n,&k,&m)!=EOF)
   {
      for(i=1;i<=n;i++)
        for(j=1; j<=k; j++)
          scanf("%d",&b[i][j]);
      for(i=0; i<=n; i++)
         x[i]=2;
      sum=0;
      m-=n;
      for(i=1;i<=m;i++)
        sum+=fun();
      for(i=1;i<=n; i++)
        sum+=b[i][1];
       printf("%d\n",sum);
   }
}
```

****

****

N 皇后问题

```
#include <iostream>
using namespace std;
bool place(int x[],int k)
{
    for(int i=1;i<k;i++)

if((x[i]==x[k])||(abs(x[i]-x[k])==abs(i-k)))
            return 0;
        return 1;
}
void queen(int n,int x[])
{
    int k=1;
    long num=0;
    x[1]=0;
    while(k>0)
```

```
	{
		x[k]+=1;
		while((x[k]<=n)&&(!place(x,k)))
			x[k]+=1;
		if(x[k]<=n)
			if(k==n)
			{
				num++;
			}
			else
				x[++k]=0;
			else
				x[k--]=0;
	}
	cout<<num<<endl;
	return;
}

int main()
{
	int n;
	cin>>n;
	int x[100];
	x[0]=0;
	queen(n,x);
	return 0;
}
/*************************************************************************/
```

/* 最优三角形问题

```
*/
/*time @    2010.11.27                                  */
/*author @    mengqinduan
        */
/*s[i][j]=(1)max(s[i-1][j-1],s[i-1][j])+a[i][j]    0<j<i;        */
/*(2)=  s[i-1][0]  +  a[i][j]            j=0    */
/*(3)=s[i-1][j-1] +a[i][j]        j=i                    */
/*************************************************************************/
#include<stdio.h>
#define    MAX 10000
int a[MAX][MAX],s[MAX][MAX];
int max(int a, int b)
{
	return a > b ? a : b;
}
int main()
{
	int n,    i, j;
	scanf("%d",&n);//三角形的大小
	for (i=0; i<n; i++)
	{
		for (j=0; j<=i; j++)
		{
			scanf("%d",&a[i][j]);
		}
	}
	s[0][0] = a[0][0];
	for (i=1; i<n; i++)
	{
		for (j=0; j<=i; j++)
		{
			if (j==0)
			{
				s[i][j] = s[i-1][0] + a[i][j];
			}
			else if (j == i)
			{
				s[i][j]  =  s[i-1][j-1]  +  a[i][j];
			}
			else
			{
				s[i][j]  =  max(s[i-1][j-1], s[i-1][j])+a[i][j];
			}


		}
	}
	int tmp = 0;
	for (i=n-1,j=0; j<n; j++)
```

```
        {
                if (s[i][j] > tmp)
                {
                        tmp = s[i][j];
                }
        }
        printf("%d\n", tmp);
        return 0;
}
```

****

动归解背包问题

```
# include<string.h>
#include<iostream>
using namespace std;
# define  N    100
int m[N][N];
int w[N],v[N],best[N];
int max(int a,int b)
{
        return a>b?a:b;
}
int   main()
{
        int n,c;
        cin>>n>>c;//输入背包的容量跟物品的件数
        int i,j,k;
        memset(m,0,sizeof(m));
        memset(w,0,sizeof(w));//背包物品的重量
        memset(v,0,sizeof(v));//背包物品的价值
        memset(best,0,sizeof(best));//记录物品有没有选
        for(i=1;i<=n;i++)
            cin>>v[i];
        for(i=1;i<=n;i++)
              cin>>w[i];
        for(i=1;i<=n;i++)
          for(j=1;j<=c;j++)
          {
                if(j>=w[i])
m[i][j]=max(m[i-1][j],m[i-1][j-w[i]]+v[i]);
                else
                m[i][j]=m[i-1][j];
          }
        i=n;
        j=c;
        /*从最优解开始寻找，如果物品包含件数多的跟少一件的价值一样
        说明最后的那件物品没有装进去，如果大的话，最后那件物品装进去了
，接下来看是，装了该物品之后剩余空间所装下的物品，以此循环，最终得到解。*/
        while(i)//求解哪些物品被选择了
        {
            if(m[i][j]>m[i-1][j])
            {
                best[i]=1;
                j-=w[i];
                i--;
            }
            else
            {
                  best[i]=0;
                  i--;
            }
        }
        cout<<"Optimal value is\n"<<m[n][c]<<endl;
        for(i=1;i<=n;i++)
        cout<<best[i]<<" ";
        cout<<endl;
        system("pause");
        return 0;
}
```

****
****
****

会场安排：

```
#include<iostream>
#include<algorithm>
using namespace std;
#define max 10000
typedef struct node
{
   int end;
   int start;
}good;
good a[max];
int b[max];
int bigger(good a,good b)
{
    return a.start<b.start;
}
int  greedy(int n)
{
        int k,j;
        int count=0;
     sort(a,a+n,bigger);
         int i,tag;
         for (i=0;i<n;i++)
         {
          tag=a[i].end;
          if(!b[i])
          {
             for(j=i+1;j<n;j++)
             if(!b[j])
             {
                  if(a[j].start>=
tag)
                  {
                     tag=a[j].end
;
                     b[j]=1;
                  }
             }
          }
         }
         for(i=0;i<n;i++)
         {
        if(!b[i])
        count++;
        }
         return count;
}

int  main()
{
        int n,c;
        while(scanf("%d",&n)!=EOF
)
        {
         for(int i=0;i<n;i++)
         {
          scanf("%d%d",&a[i].star
t,&a[i].end);
          b[i]=0;
          }
         printf("%d\n",greedy(n))
;
        }
        return 0;
}
```

****

统计数字问题

```
#include <iostream>
#include <math.h>
using namespace std;
long int power(int n)
{
        int sum=1;
        for(int i=0;i<n;i++)
        {
                sum=sum*10;
        }
        return sum;
}

int main()
{
        long num, temp, a[10];
        int i, j, k, w, x[10], y[10];
        for(i=0; i<10; i++)
        {
                a[i] = 0;
        }
        cin>>num;
        w = 0;
        temp = num;
        while(temp)
        {
                w++;
                temp /= 10;
        }
        temp = num;
        x[0] = temp/power(w-1);
        y[0] = temp-x[0]*power(w-1)+1;

        for(j=1; j<x[0]; j++)
        {
                a[j] += power(w-1);
        }
        a[j] += y[0];
        for(i=w-2, k=0; i>0; i--, k++)
        {
              x[k+1] = temp/power(i);

      y[k+1] = temp-x[k+1]*power(i)+1;

     for(j=0; j<x[k+1]-x[k]*10+1; j++)
                {

        a[j] += (x[k]+1)*power(i);
                }
                a[0] -= power(i);

      a[j-1] = a[j-1]-power(i)+y[k+1];
                for(;j <10; j++)
                {

        a[j] += x[k]*power(i);
                }
        }
        if(w != 1)
        {
                for(j=0; j<y[k]; j++)
                {
                     a[j] += x[k]+1;
                }
                for(; j<10; j++)
                {
                      a[j] += x[k];
                }
                a[0]--;
        }
        for(j=0; j<10; j++)
        {
                cout<<a[j]<<endl;
        }
        return 0;
}
```

****

****

字典序问题

```
#include<iostream>
#include<cstdio>
#include<string>
using namespace std;
int  c(int  m,int  n)
{
        if(n==0||m>n)
                return 0;
        double f=1.0;
        int i,t;
        if(n/2<m)t=n-m;
        else t=m;
        for(i=1;i<=t;i++)
        {
                f*=n-i+1;
                f/=i;
        }
        return (int )f;
}
int  check(string str)
{
        for(int  i=0;i<str.size()-1;
i++)
        {

        if(str[i]>=str[i+1])
                        return 0;
        }
        return 1;
}
int main()
{
        int  s[12]={0};
        int n,l,i;
```

```
         for(i=1;i<=11;i++)

         s[i]=s[i-1]+c(i,26);
         string str;
         cin>>n;
         while(n--)
         {
                  cin>>str;
                  if(!check(str))
                  {

         cout<<0<<endl;
                           continue;
                  }
                  int  r=str.size();
                  int  sum=c(r,26);

         for(i=1,l=r-1;i<=r;i++,l--)

         sum-=c(i,26-(str[l]-'a'+1));

         cout<<sum+s[r-1]<<endl;
         }
         return 0;
}
```

****

金币列阵问题

```
#include<iostream>
using namespace std;
const int N = 110 ;
int a[N][N],b[N][N],c[N][N];
int n,m,num;
void ChangeL(int x,int y)//变换列
{
    if(x==y)return;
    int i;
    for(i=1;i<=n;i++)
    {
        int temp=c[i][y];
        c[i][y]=c[i][x];
        c[i][x]=temp;
    }
    num++;
}
void ChangeH(int x)//变换行
{
    int i;
    for(i=1;i<=m;i++)
    {
      c[x][i]^=1;
    }
}

bool Same(int x,int y) //判断列是否满足条件
{
   int i;
   for(i=1;i<=n;i++)
       if(b[i][x]!=c[i][y])return false;
       return true;
}
int main()
{
    int t;
    bool found;
    while(cin>>t)
    {
        while(t--)
        {
            cin>>n>>m;
            int i,j;
            for(i=1;i<=n;i++)
                for(j=1;j<=m;j++)
                {
                   cin>>a[i][j];
                }



            for(i=1;i<=n;i++)
                for(j=1;j<=m;j++)
                    cin>>b[i][j];
            int k;
            int ans=n+m+1;
            for(k=1;k<=m;k++)//枚举各列为第一列
            {
```

```
       for(i=1;i<=n;i++)
           for(j=1;j<=m;j++)

               c[i][j]=a[i][j];

       num=0;
       ChangeL(1,k);
       for(i=1;i<=n;i++)
       {
           if(c[i][1]!=b[i][1])
//该行不满足条件
           {
               ChangeH(i);//变
换行
               num++;
           }
       }
       for(i=1;i<=m;i++)//检查
每列是否满足条件
       {
           found=false;
           if(Same(i,i))
           {
               found=true;
               continue;
           }
           for(j=i+1;j<=m;j++)
//寻找 temp 中与 b 的 i 列相同的列
           {
              if(Same(i,j))
              {
                  if(Same(j,j))
continue;
                  ChangeL(i,j);

                  found=true;

                  break;
              }
           }
           if(found==false)//
找不到该列对应列
           {
             break;
                  }
              }

             if(found==true&&num<ans)
ans=num;
            }

            if(ans<n+m+1)cout<<ans<<e
ndl;
            else
               cout<<"-1"<<endl;
        }
    }
    return 0;
}
*************************************
```

最大间隙问题：

```
#include<stdio.h>
#include<stdlib.h>

#define N 10000000
double a[N];
int Comp(const void *a, const void*b
)
{return *(double*)a>*(double*)b
?1:-1;
}
int main()
{
        int n;
        int i;
        double t;
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                scanf("%lf",&a[i]);
        }
qsort(a,n,sizeof(a[0]),Comp );
        double max=0;
        for(i=0;i<n-1;i++)
        {
         t=a[i+1]-a[i];
                if(t>max)
```

```
          {
               max=t;
          }
     }
     printf("%lf\n",max);
     return 0;
}
```

```
************************************
```

众数问题

```
#include<stdio.h>
#include<stdlib.h>
#define N  10000100
int a[N];
int Comp(const void *a,const voi
d *b)
{
    return *(int *)a>*(int *)b?1:
-1;
}
int main()
{
    int k=1,tag,i;
    int max=1;
    int num;
    scanf("%d",&num);
    for(i=0;i<num;i++)
    {
       scanf("%d",&a[i]);
    }
    qsort(a,num,sizeof(a[0]),Com
p);
    tag=a[0];
    for(i=0;i<num-1;i++)
    {
      if(a[i]==a[i+1])
      {
         k++;
      }
      else
      {
         if(k>max)
         {
            max=k;
            tag=a[i];
         }
         k=1;
      }
    }
    printf("%d\n",tag);
    printf("%d\n",max);
    return 0;
}
************************************
```

半数集问题：

```
#include<stdio.h>
#define MAX 1001
int ans[MAX];
int main()
{
    int n,i;
    ans[0]=ans[1]=1;
    for(i=2;i<1001;i+=2)
    {
      ans[i]=ans[i-2]+ans[i/2];
      ans[i+1]=ans[i];
    }
    while(scanf("%d",&n)!=EOF)
    {
       printf("%d\n",ans[n]);
    }
    return 0;
}
**********************************
**********************************
```

士兵站队问题

```
#include<iostream>
#include<algorithm>
using namespace std;
int x1[10001],x2[10001],y[10001],
;
```

```
int main()
{
    int i,n,midx,midy;
    int ans=0;
    while(cin>>n)
    {
        if(n==0)
            break;
        for(i=0;i<n;++i)
        {
           cin>>x1[i]>>y[i];
        }
        sort(x1,x1+n);//将x1数组里面的值按升序排序
        sort(y,y+n);//y数组里面的值也按升序排序
        midy=y[n>>1];//为y数组中间值
        for(i=0;i<n;++i)
        {
            x2[i]=x1[i]-i;
        }
        sort(x2,x2+n);
        midx=x2[n>>1];
        for(int i=0;i<n;++i)
        {
            ans+=abs(x2[i]-midx);
            ans+=abs(y[i]-midy);
        }
        cout<<ans<<endl;
    }
    return 0;
}
```

****
****

字符查找替换问题

```
#include <iostream>
#include <string>
using namespace std;
char a[10000],b[10],c[10];

int main()
{
        int i,n,j,k,l1,l2,l3,tag=1;
        cin>>n;
        while (n--)
        {
                cin>>a>>b>>c;
                l1=strlen(a);
                l2=strlen(b);
                for (i=0; i<l1; )
                {
                        k=0;
                        tag=1;

        for (j=i; j<l1; j++)
                        {

        if (k>=l2)
                                {

                                }

        if (a[j]!=b[k++])
                                {


                                }

                        }
                        if (k<l2)
                        {

        tag=0;
                        }
                        if (!tag)
                        {

        cout<<a[i];

        i++;
                        }
                        else
                        {
```

```
        cout<<c;

        i+=12;
                        }
                }
                cout<<endl;
        }
        return 0;
}
```

****
****

士兵站立问题

```
#include<iostream>
using namespace std;
#include <string.h>
#define max 1001
int a[max];
struct node
{
        int leftb;
        int lefts;
        int rightb;
        int rights;
}b[max];

int main()
{
        int num,n,i,j;
        scanf("%d",&num);
        while (num--)
        {
                scanf("%d",&n);
                for (i=0;i<n;i++)
                {

        scanf("%d",&a[i]);

        b[i].leftb=b[i].rights=b[
i].rightb=b[i].lefts=0;
                }

        for (i=1;i<n-1;i++)
                {

        for (j=0;j<n;j++)
                        {

        if (j<i)
        {

      a[i]<a[j]?b[i].lefts++:b[i].
leftb++;

       if (a[i]==a[j])
      {

      b[i].lefts++;
      }
       }

        if (j>i)
       {

      a[i]<a[j]?b[i].rights++:b[i]
.rightb++;

      if (a[i]==a[j])
       {

        b[i].rights++;
                                        }
                                }
                        }
                }
                int sum=0;

        for (i=1;i<n-1;i++)
                {

        sum+=b[i].leftb*b[i].righ
ts+b[i].lefts*b[i].rightb;
                }
        printf("%d\n",sum);
        }
        return 0;
}
************************************
```

寻宝问题

```
#include <iostream>
#include <algorithm>
using namespace std;
#define max 100000
struct good
{
        int w,v;//宝藏的重量跟价值
}a[max];
bool bigger(good a,good b)
{
        return a.v>b.v;
}

int main()
{
        int num,i,n;
        int  c,Value;
        scanf("%d",&num);
        while (num--)
        {

        scanf("%d%d",&n,&c);//输入宝藏的箱数跟所带背包的容量
                for (i=0;i<n;i++)
                {

        scanf("%d%d",&a[i].w,&a[i].v);
                }

        sort(a,a+n,bigger);
                Value=0;

        for (i=0;i<n&&c>0;i++)
                {
        if (a[i].w<c)
                        {

        Value+=a[i].v*a[i].w;

        c-=a[i].w;
                        }
                        else
                        {

        Value+=a[i].v*c;

        break;
                        }
                }

        printf("%d\n",Value);
        }
        return 0;
}
************************************
************************************
```

修路问题

```
#include<iostream>
using namespace std;
#define max 1001
int numVertices;
int Edge[max][max];
int lowcost[max];
int closest[max];
bool s[max];
void Prim () {
         int min;
         s[1]=true;
         for (int i=2;i<=numVertices;i++)
         {

        lowcost[i]=Edge[1][i];
                closest[i]=1;
                s[i]=false;
         }
        for (int i=1;i<numVertices;i++)
        {
                min=max;
                int j=1;
                for (int k=2;k<=numVertices;k++)
                {
                        if ((lowcost[k]<min)&&(!s[k]))
                        {
                                min=lowcost[k];
                                j=k;
                        }
                }
                s[j]=true;
                for (int k=2;k<=numVertices;k++)
                {
                        if ((Edge[j][k]<lowcost[k])&&(!s[k]))
                        {
                                lowcost[k]=Edge[j][k];
                                closest[k]=j;
                        }
                }
        }
        int sum=0;
       for(int k=2;k<=numVertices;k++)
              sum+=lowcost[k];
       cout<<sum<<endl;
}

int  main()
{
        int i,j,k,num,n,m,x;
        scanf("%d",&num);
        while(num--)
        {
       scanf("%d%d",&numVertices,&m);
       for(i=1;i<=numVertices;i++)
         for(j=1;j<=numVertices;j++)
           {
               Edge[i][j]=max;
               if(i==j)
               Edge[i][j]=0;
           }
       for(i=1;i<=m;i++)
       {
          scanf("%d%d%d",&j,&k,&x);
          Edge[j][k]=Edge[k][j]=x;
       }
       Prim();
        }
        return 0;
}
```

****

选课问题

```
#include<stdio.h>
#include<string.h>
int m[2000001];
int  main()
{
        int n,c,i,j,w,v;;
        while(scanf("%d%d",&n,&c)!=EOF)
        {

                for(i=0;i<=c;i++)
                m[i]=0;
                for(i=0;i<n;i++)
                {
```

```
         scanf("%d%d",&v,&w);

         for(j=c;j-w>0;j--)
                         {

         if(m[j-w]!=0&&m[j-w]+v>m[j]

         m[j]=m[j-w]+v;
                         }
                         if(m[j]<v)
                         m[j]=v;
        }
        v=0;
        for(i=0;i<=c;i++)
        v=v<m[i]?m[i]:v;
        printf("%d\n",v);
         }
         return 0;
}
```

****
****

拯救公主问题

```
#include <iostream>
using namespace std;
#define MAX 21
int a[MAX][MAX];
int max(int a,int b)
{
        return a>b?a:b;
}
int main()
{
        int num,m,n,i,j;
        scanf("%d",&num);
        while (num--)
        {
```

```
                scanf("%d%d",&n,&m);
                for (i=1;i<=n;i++)
                        {

                for (j=1;j<=m;j++)
                                {

                scanf("%d",&a[i][j]);
                                }
                        }
                for (i=1;i<=n;i++)
                        {

                for (j=1;j<=m;j++)
                                {

                if (i==1&&j==1)
                                          {
                                          }

                else if (i==1)
                {

               a[i][j]=a[i][j-1]-a[i][j];
                }

                else if (j==1)
               {

             a[i][j]=a[i-1][j]-a[i][j];
                      }

                else
               {

        a[i][j]=max(a[i-1][j],a[i][j-
1])-a[i][j];
                }


                                }
                        }
                        if (a[n][m]>0)
                        {
```

```
        printf("%d\n",a[n][m]);
                }
                else
                {

        printf("bad luck!\n");
                }
        }
        return 0;
}
```

****
****

最小机器重量问题

```
#include<iostream>
using namespace std;
int n,m,c;
int weight[100][100]={0},cost[100][100]={0},best[100],x[100];
int bestw=1000000,cw,cc;
void backtrack(int i)
{
        if(i>n)
        {
                if(cw<bestw)
                {
                        bestw=cw;

        for(int j=1;j<=n;j++)

        best[j]=x[j];
                }
                return ;
        }
        for(int j=1;j<=m;j++)
        {
                x[i]=j;
                cw+=weight[i][j];
                cc+=cost[i][j];

        if(cc<=c&&cw<bestw)backtrack(i+1);
                cw-=weight[i][j];
                cc-=cost[i][j];
        }

}
int main(){
        int i,j;
        while(cin>>n>>m>>c)
        {

        bestw=1000000,cw=cc=0;
                for(i=1;i<=n;i++)

        for(j=1;j<=m;j++)

        cin>>cost[i][j];
                for(i=1;i<=n;i++)

        for(j=1;j<=m;j++)

        cin>>weight[i][j];
                backtrack(1);
        if(bestw!=1000000)

        cout<<bestw<<endl;
                else

        cout<<"No Solution!"<<endl;
        }
        return 0;
}
```

****
****

战车问题

```
#include <iostream>
#include <ctime>
#include <cerrno>
using namespace std;
const unsigned long maxshort = 65536L;
const unsigned long multiplier = 1194211693L;
const unsigned long adder = 12345L;
int* state;
int** link;
char** row;
class RandomNumber
{
private:
unsigned long randSeed;
public :
RandomNumber(unsigned long s =0 );
unsigned short Random(unsigned long n);
double fRandom(void);
};
RandomNumber::RandomNumber(unsigned long s)
{
if(s==0)randSeed = time(0);
else randSeed = s;
}
unsigned short RandomNumber::Random(unsigned long n)
{
randSeed = multiplier * randSeed + adder;
return (unsigned short)((randSeed>>16)%n);
}
double RandomNumber::fRandom(void)
{
return Random(maxshort)/double(maxshort);
}
bool MakeRow(char** &row,int rows,int cols)
{
row=new char* [rows];
for(int i=0;i<rows;i++)
row[i] = new char[cols];
return true;
}
```

```
bool Make2DArray(int** &link,int rows,int cols)
{

link=new int* [rows];
for(int i=0;i<rows;i++)
link[i] = new int[cols];
return true;
}
void init(int n)
{
int i,j,k,x,y;

state=new int[n*n+2];
Make2DArray(link,n*n+1,2*n+1);
state[0]=-1;
state[n*n+1]=-1;
for(int no=1;no<=n*n;no++){
i=(no-1)/n;j=(no-1)%n;
link[no][0]=0;state[no]=-1;
if(row[i][j]=='.'){
state[no]=0;k=0;y=j;
while((y<n-1)&&(row[i][y+1]=='.')){
link[no][0]++;y++;k++;
link[no][k]=i*n+y+1;
}
y=j;
while((y>0)&&(row[i][y-1]=='.'))
{
link[no][0]++;y--;k++;
link[no][k]=i*n+y+1;
}
x=i;
while((x<n-1)&&(row[x+1][j]=='.')){
link[no][0]++;x++;k++;
link[no][k]=x*n+j+1;
}
x=i;
while((x>0)&&(row[x-1][j]=='.'))
{
link[no][0]++;x--;k++;
link[no][k]=x*n+j+1;
}
}
}
}
int main()
{
int n,i,j;
RandomNumber rnd;
cin>>n;
MakeRow(row,n*n+1,2*n+1);
for(i=0;i<n;i++)
for(j=0;j<n;j++)
cin>>row[i][j];
init(n);

int max=0,rept=0,put=0;
while(rept<100000){
rept++;int count=0;
while(true){
int x=rnd.Random(n*n)+1,c=x;
while((x<=n*n)&&(state[x]!=put))
x++;
if(state[x]!=put){
x=c;
while((x>0)&&(state[x]!=put))x--;
}
if(state[x]==put){
count++;
for(i=1;i<=link[x][0];i++)
if(state[link[x][i]]==put)state[link[x][i]]++;
state[x]++;
}
else
break;
}
if(count>max)max=count;
put++;
}
cout<<max<<endl;
return 0;
}
```

****

****

最大字段和问题

```
#include <iostream>
using namespace std;
#define max 100000
int a[max];
int MaxSum(int n, int *a)
{
        int sum=0,b=0,i;
        for (i=0;i<n;i++)
        {
                if (b>0)
                {
                        b+=a[i];
                }
                else
                {
                        b=a[i];
                }
                if (b>sum)
                {
                        sum=b;
                }
        }
        return sum;
}

int main()
{
        int n,i;
        cin>>n;
        for (i=0;i<n;i++)
        {
                cin>>a[i];
        }
        cout<<MaxSum(n,a)<<endl;
        return 0;
}
```

****

****

最优合并问题

```
#include <algorithm>
```

```
#include <iostream>
using namespace std;
#define N 10002
int a[N],b[N];
int max(int data[],int k) //最多
次数
{
        int i;
        int sum=0;int sum1=0;
        data[k]=data[k-1];
        for(i=k-1;i>=1;i--)
        {

        sum=data[k]+data[i-1];
                data[k]=sum;
        sum1=sum1+(sum-1);
        }
        return sum1;
}

int main()
{
        int n;
        int i,k,t,tag,r,j,sum,sum
1;
        while(scanf("%d",&n)!=EOF
)
        {
                sum=0;
                sum1=0;
                for(i=0;i<n;i++)
                {

        scanf("%d",&a[i]);

        b[i]=20001;
                }
                sort(a,a+n);

        a[n]=a[n+1]=1073741800;

        for (k=r=j=i=0;k<n-1;k++)
                {
```

```
        b[k]=a[i]+a[i+1];
                        tag=0;

        if ( b[k]>(t=a[i]+b[j]))
                        {

        b[k]=t;

        tag=1;
                        }

        if ( b[k]>(t=b[j]+b[j+1])
)
                        {

        b[k]=t;

        tag=2;
                        }
                        if (!tag)
                        {

        i+=2;
                        }

        else if (tag==1)
                        {

        i++;

        j++;
                        }
                        else
                        {

        j+=2;
                        }

        sum+=b[k]-1;
                }
                sum1=max(a,n);
        printf("%d %d\n",sum1,sum)
;
        }
        return 0;
}
************************************
************************************
```

最佳调度问题（回溯）

```
#include <iostream>
using namespace std;
#define max 10000
int k,n,best;
int len[max],t[max];
int comp()
{
        int tmp=0,i;
        for (i=0;i<k;i++)
        {
                if (len[i]>tmp)
                {

        tmp=len[i];
                }
        }
        return tmp;
}

void traceback(int dep)
{
        int i;
        if (dep>n)
        {
                int tmp=comp();
                if (tmp<best)
                {
                        best=tmp;
                }
                return;
        }
        for (i=0;i<k;i++)
        {
                len[i]+=t[dep];
                if (len[i]<best)
                {

        traceback(dep+1);
                }
                len[i]-=t[dep];
        }
}

int main()
{
        int i;
        while(scanf("%d%d",&n,&k)!
=EOF)
        {
                best=INT_MAX;
        for (i=1;i<=n;i++)
                {

        scanf("%d",&t[i]);
                }
                for (i=0;i<k;i++)
                {
                        len[i]=0;
                }
                traceback(1);

        printf("%d\n",best);
        }
        return 0;
}
************************************
************************************
```

子集树问题

```
#include<iostream>
using namespace std;
#define max 1000000
int a[max];
int best=0,cw=0;
int n,c,tag=0,sum=0;
void traceback(int i)
{
        if(i>n)
        {
        if(cw>best&&cw<=c)
                best=cw;
                if(best==c)
                tag=1;
                return ;
        }
        if(cw+sum>best&&!tag)
        {
        if(cw+a[i]<=c)
        {
            sum-=a[i];
                cw+=a[i];
                traceback(i+1);
                cw-=a[i];
                sum+=a[i];
        }
         sum-=a[i];
        traceback(i+1);
        sum+=a[i];
   }
}
int main()
{
    int i,j,k;
        while(scanf("%d%d",&n,&c)!
=EOF)
        {
                for(i=1;i<=n;i++)
                {

        scanf("%d",&a[i]);

        sum+=a[i];
                }
        //sort(a+1,a+n+1);
                traceback(1);
                cout<<best<<endl;
                best=0;
                tag=0;
                cw=0;
                sum=0;
        }
    return 0;
}
```

****
****

子集和问题

```
#include <stdio.h>
#include <stdlib.h>

int  a[10000];
bool x[10000];//数组开小了
int sum,cur,n,dest;
bool flag=true;

void fun(int t)
{
   if(cur==0)
     flag=false;
   if(t<=n && flag && cur>0)
   {
      if(cur<sum)
      {
         sum-=a[t];
         cur-=a[t];
         x[t]=true;
         fun(t+1);
         cur+=a[t];
         sum+=a[t];
         if(flag)
         {
            sum-=a[t];
            x[t]=false;
            fun(t+1);
            sum+=a[t];
         }
      }
      else if(cur==sum)
      {
         flag=false;
         for(t;t<=n;t++)
          x[t]=true;
      }
   }
}
```

```
int main()
{
    int i;
   while(scanf("%d%d",&n,&cur)!=
EOF)
   {
      flag=true;//忘记了
      for(i=1;i<=n;i++)
       x[i]=false;
         sum=0;
         cur;
        for(i=1; i<=n; i++)
          {
             scanf("%d",&a[i]);
             sum+=a[i];
           }
        i=1;
       fun(i);
   if(flag!=true)
   {
       for(i=1; i<=n; i++)
         if(x[i])
           break;
       printf("%d",a[i]);
       for(++i; i<=n; i++)
         if(x[i])
        printf(" %d",a[i]);
      printf("\n");
   }
   else
   printf("No Solution!\n");

   }
   return 0;
}
```

****
****

幸运数字问题

```
#include <iostream>
using namespace std;
#define  max 10000
int s[max];
int main()
{
        int n,a,tag,i,j;
        while(scanf("%d%d",&n,&a)!
=EOF)
        {
                tag=0;
                for (i=0;i<n;i++)
                {

        scanf("%d",&s[i]);
                }
                for (i=0;i<n;i++)
                {

        for (j=i;j<n;j++)
                        {

        if (s[i]*s[j]==a)
                                {
                                        tag=1;
                                        printf("YE
                                        break;
                                }
                        }
                        if (tag)
                        {

        break;
                        }
                }
                if (!tag)
                {

        printf("NO\n");
                }
        }
        return 0;
}
```

****
****

最小权顶点覆盖

```
#include<iostream>
using namespace std;
```

```
#define MIN 100000
int m,n,u,v,best;
int a[100][100],w[100],c[100];
int place()
{
        int i,j,t;
        i=1;
        while (i<=n)
        {
                t=0;
                if(c[i]==0)
                {
                        j=1;

        while(j<i)
                        {

        if(a[j][i]==1&&c[j]==1)


        j++;
                        }
                        j++;

        while(j<=n)
                        {

        if(a[i][j]==1&&c[j]==1)


        j++;
                        }
                        if(t==0)

        return 0;
                }
                i++;
        }
        return 1;
}
void BackTrack(int i,int s)
{
        if(s>=best)
                return;
        if(i>n)
        {
                if(place())
                        best=s;
                return;
        }
        c[i]=0;
        BackTrack(i+1,s);
        c[i]=1;
        BackTrack(i+1,s+w[i]);
}

int main()
{
        int i,j,k;
        scanf("%d%d",&n,&m);
        for(i=1;i<=n;i++)
        {
        scanf("%d",&w[i]);
                c[i]=0;

        }
        for(i=1;i<=n;i++)
                for(j=1;j<=n;j++)

        a[i][j]=0;
        for(k=1;k<=m;k++)
        {

        scanf("%d%d",&u,&v);
                a[u][v]=1;
        }
        best=MIN;
        BackTrack(1,0);
        printf("%d\n",best);
        return 0;
}
```

****
****

0/1 背包问题

```
#include<iostream>
using namespace std;
#define N 100
```

```
double limitw,tolv,maxv;
int option[N],cop[N];
struct
{
        double w;
        double v;
}a[N];
int n;   //物品数量
void find(int i,double tw,double
 tv)
{
        int k;
        if(tw+a[i].w<=limitw)
         // 包含物品 i 是可接受的
        {
        cop[i]=1;
        if(i<n-1) find(i+1,tw+a[i]
.w,tv);
        else
        {
                for(k=0;k<n;++k)

          option[k]=cop[k];
                maxv=tv;
        }
        cop[i]=0;
        }
        if(tv-a[i].v>maxv)
       // 不包含物品 i 仅是可考虑
的
        if(i<n-1) find(i+1,tw,tv-
a[i].v);
        else
        {
           for(k=0;k<n;++k)
         option[k]=cop[k];
           maxv=tv-a[i].v;
        }
}
int main()
{
        int k;
        double w,v;
        tolv=0.0;
        cin>>n>>limitw;
        for(k=0;k<n;k++)
        {
           cin>>a[k].v;
           tolv+=a[k].v;
        }
        for(k=0;k<n;++k)
        {
           cin>>a[k].w;
        }
        maxv=0.0;
        for(k=0;k<n;++k) cop[k]=0
;
        find(0,0.0,tolv);
        cout<<"Optimal value is\n
"<<maxv<<endl;
        for(k=0;k<n;++k)

        cout<<option[k]<<" ";
//从 1 开始计数
        cout<<endl;
                return 0;

}
************************************
************************************
```

工作分配问题

```
#include<iostream>
using namespace std;
#define NUM 21
int c[NUM][NUM],best[NUM];
int answer =2147483647,n;
bool Place(int k)
{
  for (int j=1;j<k;j++)
    if (best[k]==best[j])
     return false;
  return true;
}
int bound(int k)
{
    int temp = 0;
    for(int j=1;j<=k;j++)
    {
        temp+=c[j][best[j]];
    }
    return temp;
}
void Backtrack(int t)
{
     int sum=0;
    if (t>n)
    {
        for(int j = 1;j<=n;j++)
            sum+=c[j][best[j]];
        if(sum<answer)
            answer = sum;
        return;
    }
    else
    {
        for (int i=1;i<=n;i++){
            best[t]=i;
            if (Place(t)&&bound(
t)<answer) Backtrack(t+1);
      }
    }
 }
int main()
{
    scanf(″%d″,&n);
    int i,j;
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            scanf(″%d″,&c[i][j])
;
        }
    }
    for(i=0;i<=n;i++)
        best[i] = 0;
    Backtrack(1);
    printf(″%d\n″,answer);
    return 0;
}
```

****

****

删除数问题

```
#include<stdio.h>
#include<string.h>
void find()
{
    char N[200];
    int s,i=0,j,temp,m;
    scanf(″%s″,N);
    scanf(″%d″,&s);
        m=s;
        temp=strlen(N);
    while(s>0)
    {
                i=0;

        while(i<temp && N[i]<=N[i
+1])
                {
                        i++;
                }

        for(j=i;j<temp;j++)
                {

        N[j]=N[j+1];
                }
                s--;
    }
        i=0;
        while (N[i]==’0’)
        {

        for (j=i;j<=temp-m;j++)
                {

        N[j]=N[j+1];
                }
        }
    printf(″%s\n″,N);

}
```

```
int main()
{
    find();
        return 0;
}
*********************************
*********************************
```

汽车加油问题

```
#include<iostream>
using namespace std;
int d[10000];
int main()
{
        int n,k;
        cin>>n>>k;
        int i,temp;
        int count=0;
        int flag=1;
        for(i=0; i<=k; i++)
        {
                cin>>d[i];
        }
    temp=n;
        for(i=0; i<=k; i++)
        {
                if(d[i]>n)
                {

        cout<<"No Solution"<<endl
;
                        flag=0;
                        break;
                }
                else
                {

        if(count>k)
                        {

        cout<<"No Solution"<<endl
;

        flag=0;
        break;
                        }

        if(d[i]>temp)
                        {

        count++;

        temp=n;

        temp -= d[i];
                        }
                        else
                        {

        temp -= d[i];
                        }
                }
        }
        if(flag!=0)
        {
        cout<<count<<endl;
        }
        return 0;
}
*********************************
*********************************
```

最优服务次序

```
#include<stdio.h>
#include<stdlib.h>
#define N 10000
int cmp ( const void *a , const
void *b )
{
return *(int *)a - *(int *)b;
}

int  main()
{
        int a[N],i,n;
        double t,sum=0;
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
        scanf(″%d″,&a[i]);
        }
        qsort(a,n,sizeof(a[0]),cm
p);
    for(i=0;i<n;i++)
        {
         sum+=a[i]*(n-i);
        }
    t=sum/n;
    printf(″%.2lf\n″,t);
    return 0;
}
```

****

****

程序存储问题

```
#include <iostream>
#include <algorithm>
using namespace std;
#define MAX 1000000
int a[MAX];
int main()
{
        int  n,m;
        int  i;
        int sum=0;
        scanf(″%d %d″,&n,&m);
        for (i=0;i<n;i++)
        {
        scanf(″%d″,&a[i]);
                sum+=a[i];
        }
        sort(a,a+n);
        i=n-1;
        while(sum>m)
        {
                sum-=a[i--];
        }
        printf(″%d\n″,i+1);
        return 0;
}
```

****

****

字符串比较问题

```
#include <iostream>
using namespace  std;
#define max 10000
int val[max][max];
char str1[max],str2[max];
int k;
int dist(char s1,char s2)
{
        if (s2==’ ’)
        {
                return k;
        }
        else
        {

        return abs((s1-’0’)-(s2-’
0’));
        }
}
int comp()
{
        int i,j,tmp,len1,len2;
        val[0][0]=0;
        len1=strlen(str1);
        len2=strlen(str2);
        for (i=0;i<=len1;i++)
        {

        for (j=0;j<=len2;j++)
                {
                        if (i+j)
                        {

        val[i][j]=INT_MAX;

        if ((i*j)&&(tmp=val[i-1][
j-1]+dist(str1[i-1],str2[j-1]))<
val[i][j])
                                {
                val[i][j]=tmp;
                                }
        if (i&&(tmp=val[i-1][j]+d
ist(str1[i-1],’ ’))<val[i][j])
                                {
                val[i][j]=tmp;
                                }

        if (j&&(tmp=val[i][j-1]+d
ist(str1[j-1],’ ’))<val[i][j])
                                {
                val[i][j]=tmp;
                                }
                        }
                }
        }
        return val[len1][len2];
}
int main()
{
        cin>>str1>>str2>>k;
        cout<<comp()<<endl;
        return 0;
}
```

****
****

编辑距离问题

```
#include <iostream>
#include <string>
using namespace std;
int d[1000];
int main()
{
    string A1, A2;
    int x, y, z, del;
    cin >> A1 >> A2;
    int m = A1.length();
    int n = A2.length();
    int *d = new int [n + 1];
    int i ;
    for( i = 1; i <= n; i++)
            d[i] = i;
    for( i = 1; i <= m; i++)
    {
                y = i - 1;

        for(int j = 1; j <= n; j+
+)
                {
                        x = y;
                        y = d[j];

        z = j > 1 ? d[j - 1]:i;

        del = A1[i - 1] == A2[j -
 1] ? 0:1;

        d[j] = x + del;

        if (d[j] > y + 1)  d[j] =
 y + 1;

        if (d[j] > z + 1)  d[j] =
 z + 1;
                }
    }
    cout << d[n] << endl;
    return 0;
}
```

****
****

最少硬币问题

```
#include <iostream>
using namespace std;


int T[11],Coins[11],n;
int c[20002],num[20002];
int main()
{
        int i,j,m;
        while(scanf("%d",&n)!=EOF)

        {
                for(i=0;i<n;++i)
        scanf("%d%d",&T[i],&Coins
[i]);
                scanf("%d",&m);
                for(i=1;i<=m;++i)

        c[i]=0xfffffff;
                c[0]=0;
                for(i=0;i<n;++i)
                {

        for(j=0;j<=m;++j)

        num[j]=0;

        for(j=0;j<=m-T[i];++j)

        if(num[j]<Coins[i]&&c[j]+
1<c[j+T[i]])
                                {


                                }
                }

        if(c[m]!=0xfffffff)

        printf("%d\n",c[m]);
                else

        printf("-1\n");
        }

        return 0;

}
```

****
****

独立任务最优调度

```
#include <iostream>
#include <cstdio>
#include <algorithm>
using namespace std;
```

```
//最大任务数量
#define MAXN 201
//全部在 A 机器上运行的最大时间上
限
#define MAXTIME 10001

int p[MAXTIME][MAXN];

int ai[MAXN], bi[MAXN];

int n;

int task(int n) {
        int sum = 0;
        for(int i = 0; i <= sum;
++i) {
                p[i][0] = 0;
        }
        for(int k = 1; k <= n; ++
k) {
                sum += ai[k];

        for(int j = 0; j <= sum;
++j) {

        p[j][k] = p[j][k - 1] + b
i[k];

        if(j >= ai[k]) {

        p[j][k] = min(p[j][k], p[
j - ai[k]][k - 1]);
                        }
                }
        }
        int ret = sum + 2;
        for(int i = 1; i <= sum;
++i) {

        ret = min(ret, max(p[i][n]
, i));
        }
        return ret;
}

int main() {
        scanf("%d", &n);
        for(int i = 1; i <= n; ++
i) {

        scanf("%d", &ai[i]);
        }
        for(int i = 1; i <= n; ++
i) {

        scanf("%d", &bi[i]);
        }



        int ans = task(n);
        printf("%d\n", ans);
        return 0;
}
```

****
****

最长子序列

```
#include<stdio.h>
#define  MAX 10000
int a[MAX],b[MAX];
/*b 数组记录的是按序号排列数递增
序列的最大长度*/
int main()
{
        int m, i, j, k;
        scanf("%d",&m);//序列长度
        for (i=0; i<m; i++)
        {

        scanf("%d",&a[i]);//序列
的各个数值
        }
        for (i=1, b[0]=1; i<m; i+
+)
        {

        for (j=0, k=0; j<i; j++)
                {

        if (a[j] <= a[i] && k<b[j]
)
                        {

        k = b[j];
                        }

        b[i] = k+1;
                }
        }
        int max = 0;
        for (i=0; i<m; i++)
        {
                if (b[i] > max)
                {

        max = b[i];
                }
        }
        printf("%d\n", max);
        return 0;
}
```

****
****

集合划分

```
#include<stdio.h>
#define MAX 100
__int64  a[MAX][MAX];

__int64  fun(__int64 n,__int64 m
)
{
   if(m==n||m==1)
   {
     a[n][m]=1;
     return 1;
   }
   a[n-1][m-1]=fun(n-1,m-1);
   a[n-1][m]= fun(n-1,m);
   a[n][m]=a[n-1][m-1]+m*a[n-1][
m];
```

```
   return a[n][m];
}

int main()
{
    __int64  n,i;
    __int64 sum=0;
    while(scanf("%I64d",&n)!=EOF
)
    {
        for(i=1;i<=n;i++)
        sum+=fun(n,i);
        printf("%I64d\n",sum);
    }
    return 0;
}
```

****
****

求回文数的个数

```
#include <stdio.h>
bool huiwen(long n)
{
        long i=n,m=0;
        while(i)
        {
       m=m*10+i%10;
       i=i/10;
    }
    return (m==n);
}
int main()
{
        int i=0,n;
        int cnt;
        while(scanf("%d",&n)!=EOF
)
        {
                cnt = 0;

        for(i=1; i<=n; i++)
                {

        if(huiwen(i))
                        {

        cnt++;
                        }
                }

        printf("%d\n", cnt);
        }
        return 0;
}
```

****
****

简单归并问题

```
#include<iostream>
using namespace std;
#include <string.h>
#define MAX 1000000
int a[MAX],b[MAX];
int main()
{
        int i,n,j,number1,number2,
tmp;
        cin>>n;
        while (n--)
        {
                cin>>number1;

        for (i=0;i<number1;i++)
                {

        cin>>a[i];
                }
                cin>>number2;

        for (i=0;i<number2;i++)
                {

        cin>>b[i];
                }

        for (i=j=0;i<number1&&j<n
umber2;)
                {
```

```
if (a[i]<=b[j])
                {

cout<<a[i];

i++;
                }
                else
                {

cout<<b[j];

j++;
                }

cout<<″ ″;
        }
        while (j<number2)
        {

cout<<b[j];

if (j+1<number2)
                {

cout<<″ ″;
                }
                j++;
        }
        while (i<number1)
        {

cout<<a[i];

if (i+1<number1)
                {

cout<<″ ″;
                }
                i++;
        }
        cout<<endl;
}
```

```
        return 0;
}
************************************
************************************
```

Fibonacci Number 问题

```
#include<stdio.h>
#define maxsize 46
int main()
 {
     int n,a,i,b[maxsize];
     scanf(″%d″,&n);
     while(n--)
     {
     scanf(″%d″,&a);
     if(a>=0&&a<=45)
     for(i=0;i<=a;i++)
     if(i<2)
        b[i]=i;
     else
        b[i]=b[i-1]+b[i-2];
     printf(″%d\n″,b[i-1]);
     }
     return 0;
 }
************************************
************************************
```

检查金币

```
#include<stdio.h>
#include<math.h>
int a[60000][10];
int b[10];
#define MIN 265716
int main()
{
        int n, i, j, min, k, tota
l;

        for(i=0; i<10; i++)
        {
                a[0][i] = 9;
        }
```

```
     for(i=0; i<10; i++)
     {

     b[i] = (int)pow(3, i);
     }

     for(i=1; i<=59048; i++)
     {
          total = i+MIN;
          min = MIN;


     for(k=0; k<10; k++)
          {

     a[i][k] = 9;
          }



     for(j=9; j>=0; j--)
          {

     if((2*b[j]+min) <= total)
               {

     a[i][j] = 11;

     if((2*b[j]+min) == total)
                    {
               break;
                    }
          min += b[j]*2;
               }

     else if((b[j]+min) <= tot
al)
               {

     a[i][j] = 10;

     if((b[j]+min) == total)
                    {

                    }
     min += b[j];
               }
          }
     }
     while(scanf(″%d″, &n) !=
EOF)
     {

     printf(″%d″, a[n-MIN][0])
;

     for(i=1; i<10; i++)
          {

     printf(″ %d″, a[n-MIN][i])
;
          }
          printf(″\n″);
     }
     return 0;
}
```

```
******************************
******************************
```

密码破译

```
#include <stdio.h>
int a[5000];
int  main()
{
   int n,m,i,j,tag;
   scanf(″%d%d″,&n,&m);
   for(i=0;i<n;i++)
   scanf(″%d″,&a[i]);
   for(i=0;i<n;i++)
   {tag=0;
   for(j=i+1;j<n;j++)
   if(a[i]+a[j]==m)
   {
      tag=1;
      break;
   }
   if(tag)
   break;
}
if(tag)
printf("%d\n%d\n",i+1,j+1);
else
printf("0\n");    return 0;
}
*************************************
*************************************
```

最大公约数

```
#include<stdio.h>
int yueshu(int a,int b)
{
   int t;
      t=a%b;
   while(t)
   {
      a=b;
      b=t;
      t=a%b;

   }
   return b;
}
int main()
{
   int number;
   scanf("%d",&number);
   int a,b;
   while(number--)
   {
   scanf("%d %d",&a,&b);
   printf("%d\n",yueshu(a,b));
   }
  return 0;
}
*************************************
*************************************
```

Faulty Odometer

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int power9(int n)
{
    int i;
    int sum=1;
    for(i = 0; i < n; i++)
    {
        sum=sum*9;
    }
    return sum;
}
int power10(int n)
{
    int i;
    int sum=1;
    for(i = 0; i < n; i++)
    {
        sum=sum*10;
    }
    return sum;
}
int main()
{
    char num[20];
    int sum;
    int i=0,j=0;
    int length;
    int N;
    scanf("%s", num);
    N = atoi(num);
    while(N)
    {
        sum=0;
        length = strlen(num);
        for (i = length-1; i > 0;
 i--)
        {
            for(j = 0; j < i;j++
)
            {
                sum=sum+( power9
(i-j-1)*(num[j]-’0’-((num[j]-’0’)
>3))+((num[i-j]-’0’)/4&&!j?1:0))
*power10(length-1-i);
            }
        }
        sum += ((num[i]-'0')>4)*
power10(length-1);
        printf("%s: %d\n", num,
N-sum);
        scanf("%s", num);
        N = atoi(num);
    }
   return 0;
}
*********************************
*********************************
```

Jolly jumper

```
#include<stdio.h>
#include<stdlib.h>
#define N 3000
int a[N],b[N];
int main()
{
        int n,i,j,k;
        while(scanf("%d",&n)!=EOF
)
        {

                for(i=0;i<n;i++)
                {

        scanf("%d",&a[i]);
                        b[i]=0;
                }
                if(n==1)
                {

        printf("Jolly\n");
                }
                else
                {

        for(j=1;j<n;j++)
                        {

        k=abs(a[j]-a[j-1]);

        b[k]=1;
                        }

        for(i=1;i<n;i++)
                        {

        if(!b[i])

        {

        printf("Not jolly\n");
                        break;
                                }

        if(i==n-1)
                    {

        printf("Jolly\n");
                    }
                        }
                }
        }
        return 0; }
********************************
********************************
```

Common permutation 问题

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char str1[1005];
char str2[1005];
int cmp(const void *a,const void
 *b)
{
        if((*(char*)a  - *(char*)
b)>0)
                return 1;
        return -1;
}
int main()
{

        int len1,len2,i,k,x,y;
        while(gets(str1))
        {

        gets(str2);


        len1 = strlen(str1);


        len2 = strlen(str2);


        qsort(str1,strlen(str1),s
izeof(str1[0]),cmp);

        qsort(str2,strlen(str2),s
izeof(str2[0]),cmp);


        k = (len1>len2)?len1:len2;


        x=0,y=0;

        while((x<len1)&&(y<len2))

                {

        if(str1[x]==str2[y])


        {

        printf("%c",str1[x]);


        x++;y++;

        }

        else if(str1[x]>str2[y])


        y++;

        else

        x++;
                }
                printf("\n");
        }
        return 0;
}
```

****
****

密码锁问题

```
#include<stdio.h>
#include<string.h>
int main()
{
    char s[10];
    int a[10], length;
    int i = 0, j = 0;
    printf("Spinlock Results\n")
;
    scanf("%s",s);
    while ( s[0]!='z' )
    {
        for(i=0;i<10;i++)
        {
            a[i]=0;
        }
        length=strlen(s);
        do
        {
            for (i = 0; i < leng
th; i++)
                a[i]=(a[i]+(int)
(s[i]-'0'))%10;
            scanf("%s",s);
        }while(s[0]!='x');
        for(j=0;j<length;j++)
        {
            printf("%d",a[j]);
        }
        printf("\n");
        i=0;j=0;
        scanf("%s",s);
    }
    return 0;
}
```

****
****

Lowest bit 问题

```
#include<stdio.h>
int main()
{
        int r,i;
        unsigned long int a;
        scanf("%d",&a);
        while(a)
        {
                i=0;r=0;
                if(a)
                {
                        while(!r)
                        {

        r=a%2;

        a=a/2;

        i++;
                        }

        printf("%d\n",1<<(i-1));
                }
                scanf("%d",&a);
        }
        return 0;
}
```

****
****

SelfNumber 问题

```
#include<stdio.h>
int  main()
{
  int i,j;
  int n;
  for(i=1;i<10000;i++)
  {
      n=0;
      for(j=i-36;j<i;j++)
          if(j>0&&i==(j+j%10+(j/
10)%10+(j/100)%10+(j/1000)%10))
          {
            n=1;
            break;
          }
          if(!n)
          printf("%d\n",i);
  }
  return 0;
}
```

****
****

日历问题

```
#include<stdio.h>
int main()
{
    int s=2000;
    int M[13]={0,31,28,31,30,31,
30,31,31,30,31,30,31};
    int N[13]={0,31,29,31,30,31,
30,31,31,30,31,30,31};
    int n,i,tem,b,a,m;
    char D[7][13]={"Saturday","S
unday","Monday","Tuesday", "Wedn
esday", "Thursday", "Friday" , }
;
    while(scanf("%d",&n),n!=-1)
    {
       tem=n;
       s=2000;
       m=0;
       i=1;
       while(n>=146097)
       {
           n-=146097;
           m++;
```

```
        }
        while(n>364&&s>0)
        {
            a=(s%400==0||(s%4==0&&s%100!=0)?1:0);
            if(n>364&&a==0){n-=365;s++;a=(s%400==0||(s%4==0&&s%100!=0)?1:0);}
            if(n>365&&a==1){n-=366;s++;a=(s%400==0||(s%4==0&&s%100!=0)?1:0);}
            if(a==1&&n<366)break;
        }
        b=(s%400==0||(s%4==0&&s%100!=0)?1:0);
        if(b)
        {
          while(n>=N[i])
          {
           n-=N[i];
           i++;
          }
        }
        else
        {
              while(n>=M[i])
              {
                   n-=M[i];
                   i++;
              }
        }

        if(m==0)printf("%d",s);
        else printf("%d",400*m+s);
        if(i>9)printf("-%d",i);
        else printf("-0%d",i);
        if((n+1)>9)printf("-%d",n+1);
        else printf("-0%d",n+1);
        printf(" ");
        puts(D[tem%7]);
    }
    return 0;
}
```

****

谁那最多奖学金问题

```
#include<stdio.h>
int main()
{
    int i,j,n,qm,py,lw,prize,max=0;
    long total=0;
    char a[20],name[20],xb,gb;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
          scanf("%s %d %d %c %c %d",a,&qm,&py,&gb,&xb,&lw);
          prize=0;
          if((qm>80)&&(lw>0)) prize+=8000;
          if((qm>85)&&(py>80)) prize+=4000;
          if(qm>90) prize+=2000;
          if((qm>85)&&(xb==’Y’)) prize+=1000;
          if((py>80)&&(gb==’Y’)) prize+=850;
          total+=prize;
          if(prize>max)
          {
              max=prize;
              for(j=0;j<20;j++)
              name[j]=a[j];
          }
    }
     printf("%s\n%d\n%ld",name,max,total);
     return 0;
}
```

****

求三角形面积

```
#include<stdio.h>
```

```
#include<stdlib.h>
int main()
{
    int x[5],y[5];
        double s;
        while(scanf("%d %d %d %d
%d %d",&x[0],&y[0],&x[1],&y[1],&
x[2],&y[2])!=EOF)
        {
                x[4]=x[0]-x[1];
                y[4]=y[0]-y[1];
                x[3]=x[0]-x[2];
                y[3]=y[0]-y[2];

        s=0.5*abs(x[4]*y[3]-x[3]*
y[4]);
                printf("%f\n",s);
        }
    return 0;
}
```

****
****

分解素因子

```
#include<stdio.h>
int  main()
{
     int m,k,n;
     scanf("%d",&n);
     while(n--)
     {
         scanf("%d",&m);
         for(k=2; k<m; )
         {
              if(m%k != 0)
              {   //不能被 k 整除,
就让 k 自加 1
               k++;
              }
              else
              {    //能被 k 整除,
就让 m 等于被除后的数
               printf("%d*", k);
               m = m/k;
                 k = 2;   //让 k 重
新等于 2
                 }
             }
             printf("%d\n", k);
         }
         return 0;
}
```

****
****

阅读顺序问题

```
#include <string.h>
#include<iostream>
using namespace std;
int main()
{
    int n,i,t,tmp;
        char a[201];
    cin>>n;
        cin.get();
    while(n--)
    {
       cin.getline(a,200);
           tmp=strlen(a);
           for (i=tmp-1;i>=0;i--)
           {
                   cout<<a[i];
           }
           cout<<endl;
    }
    return 0;
}
```

****
****

宝石问题

```
#include <stdio.h>
#include <stdlib.h>

int prime[1200];

int  a[2][1010];
bool flag[2][1020];
```

```
int pr()
{
    int k,i,j;
   for(i=3; i<1001; i+=2)
     {
        if(prime[i]==0)
         {
            k=i<<1;
            for(j=i*i; j<1001; j+=k)
               prime[j]=1;
         }
     }
    for(i=4;i<1001;i+=2)
      prime[i]=1;
    prime[1]=1;//添加了这个就对了，1不是素数
}
int main()
{
    pr();
    int n,i,j,k,l,m;
    while(scanf("%d",&n)!=EOF)
    {
    i=1;
    m=1;l=0;
    for(i=0;i<=n;i++)
      {flag[0][i]=flag[1][i]=false;
        a[0][i]=a[1][i]=0;
        }

    i=1;
    while(i<=n)
    {

       for(j=1;j<=i;j++)
         {
            scanf("%d",&a[m][j]);
            if(prime[a[m][j]]==0)
              flag[m][j]=true;

           if(a[l][j]>a[l][j-1])
             {
                if(flag[l][j])
                  flag[m][j]=true;
                  a[m][j]+=a[l][j];
             }
           else if(a[l][j]<a[l][j-1])
             {
                if(flag[l][j-1])
                  flag[m][j]=true;
                a[m][j]+=a[l][j-1];
             }
           else
           {
               a[m][j]+=a[l][j];
               if(flag[l][j] || flag[l][j-1] || flag[m][j])
               flag[m][j]=true;
               else
               flag[m][j]=false;
           }
         }
         k=m;
         m=l;
         l=k;
         i++;
    }
    k=0;
    for(i=1;i<=n;i++)
      {
        if(a[l][i]>k && flag[l][i])
          k=a[l][i];
      }

    printf("%d\n",k);
}
}
*************************************
*************************************
```

惊人的兔子问题

```
#include<iostream>
using namespace std;
#define max 91
__int64 a[max];

int main()
{
        int i,n;
        scanf("%I64d",&n);
        a[1]=1;
        a[2]=1;
        while (n!=0)
        {
        for (i=3;i<=n;i++)
                {

        a[i]=a[i-1]+a[i-2];
                }

        printf("%I64d\n",a[n]);
        scanf("%I64d",&n);
        }
        return 0;
}
```