**2557: Fibonacci数列**

这题很简单，找到比输入数字大的a和比输入数字小的b，然后输出min(n-a,b-n)就行了。

|  |
| --- |
| #include <iostream>  #include<cstdio>  using namespace std;  int main(){  int N;  while (cin>>N) {  int f0=0,f1=1,f,a,b;  while (1) {  f = f0 + f1;  f0 = f1;  f1 = f;  if (f<N) {  a = N-f;  }else{  b = f-N;  break;  }  }  cout<<min(a, b)<<endl;  }  return 0;  } |

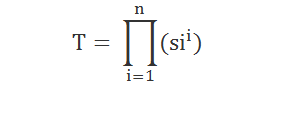
## 2558: 手机信号的显示

## 这道题需要注意小于60是E，大于等于60小于90是3G，而其他为4G，而‘-’也要对齐，这个就是简单的字符串替换，或者是模拟题。

|  |
| --- |
| #include<iostream>  #include<cstdio>  using namespace std;  char c[10][10]={  "+-----+",  "| E|",  "| |",  "| |",  "| |",  "| |",  "+-----+",  };  int main()  {  int n;  while(cin>>n)  {  for(int i=1;i<=n/20;i++)  {  for(int j=1;j<=i;j++)  c[i][j]='-';  }  if(n<60)  {  }else if(n>=60&&n<90){  c[1][4]='3';  c[1][5]='G';  }else{  c[1][4]='4';  c[1][5]='G';  }  for(int i=0;i<7;i++)  cout<<c[i]<<endl;  }  return 0;  } |

## 2559: 数位的幂

这道题不知道多少人看了，这道题没有人做，也不知道你对题目的理解程度。这道题很简单，下面看一下这道题的公式。



这个题目举个简单的例子就很好理解了。

123123 = 1^1\*2^2\*3^3\*1^4\*2^5\*3^6

= 1^5 \* 2^7 \* 3^9

这样变化就很好理解了，现在输入n位的数字，只可能出现0123456789这几个数字，统计一下每个数字是多少次幂。

方法一：快速幂计算

快速幂模板（可以log（n）时间内计算出a^n）

|  |
| --- |
| typedef long long ll;  const ll mod = 1000000007;  ll quickpw(ll a,ll b)  {  ll ans = 1,base = a;  while(b!=0)  {  if(b&1!=0)  ans=ans\*base%mod;  base=base\*base%mod;  b>>=1;  }  return ans%mod;  } |

快速幂代码

|  |
| --- |
| #include<iostream>  #include<cstdio>  #include<cstring>  using namespace std;  typedef long long ll;  const ll mod = 1000000007;  const int maxn = 1000001;  ll quickpw(ll a,ll b)  {  ll ans = 1,base = a;  while(b!=0)  {  if(b&1!=0)  ans=ans\*base%mod;  base=base\*base%mod;  b>>=1;  }  return ans%mod;  }  char c[maxn];  int main()  {  //freopen("3.in","r",stdin);  //freopen("3.out","w",stdout);  while(~scanf("%s",c))  {  ll a[11];  ll ans=1;  memset(a,0,sizeof(a));  int len = strlen(c);  for(int i=0;i<len;i++)  {  a[c[i]-'0']+=(i+1);  }  for(int i=0;i<11;i++)  {  if(a[i]){  ans=ans\*quickpw(i,a[i])%mod;  }  }  cout<<ans%mod<<endl;  }  return 0;  } |

方法二：动态规划（数位dp）

计算a^123 = a^100 \* a^20 \* a^3通过这个来计算

现在打表:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | a^1 | a^2 | a^3 | a^4 | a^5 | a^6 | a^7 | a^8 | a^9 |
| 2 | a^10 | a^20 | a^30 | a^40 | a^50 | a^60 | a^70 | a^80 | a^90 |
| 3 | a^100 | a^200 | a^300 | a^400 | a^500 | a^600 | a^700 | a^800 | a^900 |

现在可以查表计算出a^123 = (3,1)\*(2,2)\*(3,1)

同理假设a=0,1,2,3,4,5,6,7,8,9可以计算出各个结果最后再输出。

## 2560: 四边形最大面积

假设四个边依次为a，b，c，d，可以组成四边形的充要条件为a+b+c>d，其中a，b，c均小于d。

**https://img-blog.csdn.net/20130729123511890?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast**

**其中https://img-blog.csdn.net/20130729123640609?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast**

**这也即是圆内接四边形的面积公式。**

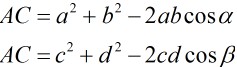
**证明过程：**

**首先我们把四边形的四个顶点和四条边顺次分别记为：A,B,C,D和a,b,c,d，角B和D分别记为https://img-blog.csdn.net/20130729124120921?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast，设面积为S**

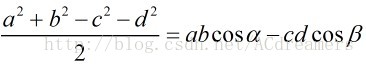
**这样就可以写推导过程了：**

**https://img-blog.csdn.net/20130729124337406?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast，亦即https://img-blog.csdn.net/20130729124502296?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast   (1)**

**对于对角线AC，由余弦定理对于https://img-blog.csdn.net/20130729124948593?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast有：**

****

**消去AC得到：**

**     (2)**

**将(1)^2+(2)^2后得到：**

**https://img-blog.csdn.net/20130729130130687?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast**

**所以到了这里可以看出要使S最大，必须使https://img-blog.csdn.net/20130729130308453?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast等于-1.**

**这样就知道了，https://img-blog.csdn.net/20130729130552531?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast，所以这就证明了，S最大时，四点都在外接圆上。**

**然后我们继续化简S，我们进一步得到：**

**https://img-blog.csdn.net/20130729131126687?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast**

**然后把后面的用平方差公式继续展开：**

**https://img-blog.csdn.net/20130729131421546?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast**

**继续平方差公式展开得到：**

**https://img-blog.csdn.net/20130729131629921?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast**

**现在我们设：https://img-blog.csdn.net/20130729131721250?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast，这样我们就得到了任意给4条边形成四边形面积最大值的公式：**

**https://img-blog.csdn.net/20130729131831500?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvQUNkcmVhbWVycw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast**

|  |
| --- |
| #include <cstdio>  #include <cstring>  #include <cmath>  #include <algorithm>  #include<stack>  #include<iostream>  using namespace std;  int main()  {  //freopen("2.in","r",stdin);  //freopen("2.out","w",stdout);  double a,b,c,d;  int t;  while(cin>>t)  {  int cnt = 1;  while(cnt<=t)  {  cin>>a>>b>>c>>d;  double e[4]= {a,b,c,d};  sort(e,e+4);  if(e[0]+e[1]+e[2]<=e[3])  {  printf("Case %d: %d\n",cnt++,-1);  continue;  }  double p = (a+b+c+d)/2;  double s = sqrt((p-a)\*(p-b)\*(p-c)\*(p-d));  printf("Case %d: %.6f\n",cnt++,s);  }  }  return 0;  } |

**同时它也是圆内接四边形的面积公式。**

## 2561: 最后一个单词的长度

这道题原本是送分题，结果变成了送命题，最后难度一降再降才让好多人过了。建议错了一次及以上的的都要去看一下oj工作原理。

|  |
| --- |
| while(~scanf(“%d”)){} |

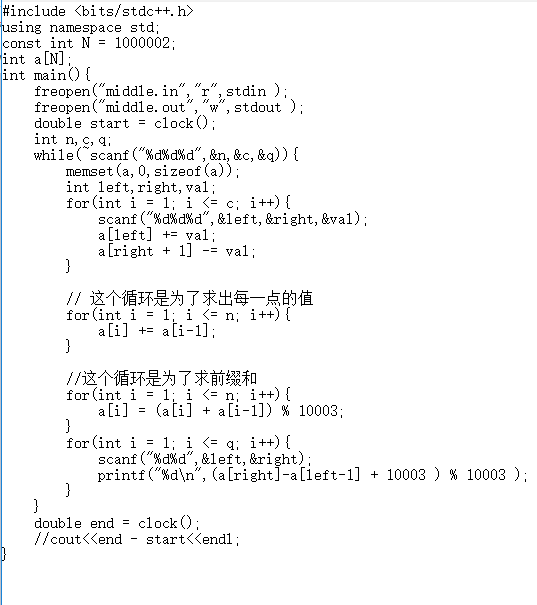
当没有输入的时候自动退出，同理，java有cin.hasNext()，c++有cin都可以判断有无输入。

代码：

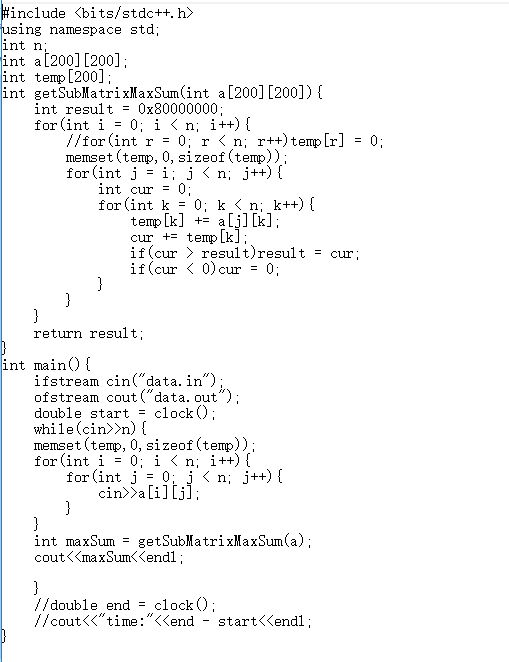
|  |
| --- |
| #include<iostream>  #include<cstring>  #include<cstdio>  char c[1001];  using namespace std;  int main()  {  int len=0;  while(cin>>c)  {  len=strlen(c);  }  cout<<len<<endl;  return 0;  } |

看到代码是不是想哭，是不是想复杂了。

**F题：**



**G题：**



**H题：**

