**Hello wold**

#include <iostream>

#include <stdio.h>

#include <string.h>

using namespace std;

int main()

{

int n;

while(scanf("%d",&n)!=EOF)

{

while(n--)

{

int num;

char str[10001];

scanf("%d",&num);

getchar();

gets(str);

int ans[500],len=strlen(str),cnt=0,now=0,cntEnd=0;

for(int i=0;i<len;i++)

{

now++;

if(str[i]=='.'||str[i]=='!'||str[i]=='?')

{

ans[cnt++]=now;

now=0;

i++;

}

}

cntEnd=cnt;

//printf("%d",cnt);

int flag=true;

//for(int i=0;i<cnt;i++)

// printf("%d ",ans[i]);

for(int i=0;i<cnt-1;i++)

{

if(ans[i]>num)

flag=false;

if(ans[i]+ans[i+1]<num)

{

ans[i+1]+=ans[i];

cntEnd--;

}

}

if(ans[cnt-1]>num)

flag=false;

if(flag)

printf("%d\n",cntEnd);

else printf("Impossible\n");

}

}

cout << "Hello world!" << endl;

return 0;

}

**特殊评价：**

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

using namespace std;

int main()

{

int n;

while(scanf("%d",&n)!=EOF)

{

while(n--)

{

double s,t,e;

scanf("%lf%lf%lf",&s,&t,&e);

if(fabs(s-t)<=e)

puts("Accept");

else puts("Wrong Answer");

}

}

cout << "Hello world!" << endl;

return 0;

}

**编辑距离**

#include <iostream>

using namespace std;

#include <stdio.h>

#include <string.h>

#include <math.h>

int main()

{

int n;

while(scanf("%d",&n)!=EOF)

{

getchar();

char a[501],b[501];

int dp[501][501];

scanf("%s%s",a,b);

int len1=strlen(a),len2=strlen(b);

for(int i=0;i<=len1;i++)

dp[i][0]=0;

for(int i=0;i<=len2;i++)

dp[0][i]=0;

for(int i=1;i<=len1;i++)

for(int j=1;j<=len2;j++)

{

if(a[i-1]!=b[j-1])

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

else

dp[i][j]=dp[i-1][j-1]+1;

}

printf("%d\n",max(len1,len2)-dp[len1][len2]);

}

// cout << "Hello world!" << endl;

return 0;

}

**查询成绩**

#include <iostream>

#include <stdio.h>

#include <algorithm>

#include <string.h>

using namespace std;

struct E{

char name[6];

int score;

bool operator < (const E &A)const

{

if(score==A.score)

{

int tmp=strcmp(name,A.name);

return tmp<0;

}

else return score<A.score;

}

}student[5001];

int main()

{

int n;

while(scanf("%d",&n)!=EOF)

{

while(n--)

{

int m;

scanf("%d",&m);

for(int i=0;i<m;i++)

scanf("%s%d",student[i].name,&student[i].score);

sort(student,student+m);

int num;

scanf("%d",&num);

while(num--)

{

int s;

scanf("%d",&s);

int i=0;

for(;i<m;i++)

{

if(s<student[i].score)

{

puts(student[i].name);

break;

}

}

if(i==m)

puts("Impossible");

}

}

}

cout << "Hello world!" << endl;

return 0;

}

**几何级数**

#include <iostream>

#include <stdio.h>

#include <math.h>

using namespace std;

int gcd(int a,int b)

{

if(b==0) return a;

else return gcd(b,a%b);

}

int lcm(int a,int b)

{

return a\*b/gcd(a,b);

}

int fun(int a,int b)//前面的数大

{

int cnt=0;

do{

a/=b;

cnt++;

}while(a%b==0);

return cnt;

}

int main()

{

int T;

while(scanf("%d",&T)!=EOF)

{

while(T--)

{

int b1,q1,n1,b2,q2,n2;

scanf("%d%d%d%d%d%d",&b1,&q1,&n1,&b2,&q2,&n2);

int cnt1=b1\*q1,cnt2=b2\*q2;

if(b1==b2&&q1==q2)

{

printf("%d\n",max(n1,n2));

}

else if(gcd(min(cnt1,cnt2),gcd(cnt1,cnt2))==min(cnt1,cnt2))

{

if(b1==q2&&b2==q1)

{

if(gcd(b1,b2)==1&&gcd(q1,q2)==1)//两两互质只有一个

printf("%d\n",n1+n2-1);

else

{

puts("buhui");//如果类似于4 2 2 2 4 2

}

}

else if(b1!=b2&&gcd(b1\*q1,b2)==b1\*q1)//第一组两个可以造出第二个数

{

if(gcd(b2/b1/q1,q1)!=q1&&b2!=b1\*q1)

{

printf("%d\n",n1+n2);

}

else if(gcd(q1,q2)!=min(q1,q2))

{

printf("%d\n",n1+n2-1);

}

else

{

if(gcd(q1,q2)==q1)

{

if(b2%b1==0)

{

int ans=fun(q2,q1),ans1=fun(b2/b1,q1);

//printf("ans=%d %d\n",ans,ans1);

int jian=0,n1S=n1-1;

if(n1S>=ans1)

{

n1S-=ans1;

jian++;

for(int i=ans,j=1;i<=n1S&&j<=n2-1;i+=ans,j++)

{

//printf("%d %d %d\n",n1S,i,ans);

jian++;

}

}

printf("%d\n",n1+n2-jian);

}

else

{

int ans=fun(q2,q1),ans1=fun(b2\*q2/b1,q1);

printf("ans=%d %d\n",ans,ans1);

int jian=0,n1S=n1-1;

if(n1S>=ans1)

{

n1S-=ans1;

jian++;

for(int i=ans,j=1;i<=n1S&&j<=n2-2;i+=ans,j++)

{

printf("%d %d %d\n",n1S,i,ans);

jian++;

}

}

printf("%d\n",n1+n2-jian);

}

}

else{

printf("%d\n",n1+n2);

}

}

}

else if(b1!=b2&&gcd(b2\*q2,b1)==b2\*q2)//第二组两个可以造出第1个数

{

if(gcd(b1/b2/q2,q2)!=q2&&b1!=b2\*q2)

{

printf("%d\n",n1+n2);

}

else if(gcd(q1,q2)!=min(q1,q2))

{

printf("%d\n",n1+n2-1);

}

else

{

if(gcd(q1,q2)==q2)

{

if(b1%b2==0)

{

int ans=fun(q1,q2),ans1=fun(b1/b2,q2);

int jian=0,n2S=n2-1;

if(n2S>=ans1)

{

n2S-=ans1;

jian++;

for(int i=ans,j=1;i<=n2S&&j<=n1-1;i+=ans,j++)

{

jian++;

}

}

printf("%d\n",n1+n2-jian);

}

else{

int ans=fun(q1,q2),ans1=fun(b1\*q1/b2,q2);

int jian=0,n2S=n2-1;

//printf("ans=%d %d\n",ans,ans1);

if(n2S>=ans1)

{

n2S-=ans1;

jian++;

for(int i=ans,j=1;i<=n2S&&j<=n1-2;i+=ans,j++)

{

jian++;

}

}

printf("%d\n",n1+n2-jian);

}

}

else{

printf("%d\n",n1+n2);

}

}

}

else if(b1==b2&&q1!=q2)

{

printf("%d\n",n1+n2-1);

}

else {

printf("%d\n",n1+n2);

}

}

else { printf("%d\n",n1+n2);}

}

}

//cout << "Hello world!" << endl;

return 0;

}

**树的最长距离（other）**

#include <iostream>

#include <string.h>

#include <vector>

#define MAX 1001

using namespace std;

int map[MAX][MAX],dis[MAX],vis[MAX],cnt[MAX],parent[MAX],path[MAX];

//dfs(起点,总节点数,是否访问过的数组,最大值,最大值下标)

void dfs(int s,int n,int vis[],int &max,int &max\_p){

vis[s] = 1;

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

if(vis[i]==0&&map[s][i]!=0)

{

dis[i] = dis[s] + map[s][i];

if(max < dis[i]){

max = dis[i];

max\_p = i;

}

parent[i] = s;

dfs(i,n,vis,max,max\_p);

}

}

}

void maxdis(int s,int n,int \_dis,int &max,int vis[]){

vis[s] = 1;

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

{

if(path[i]==0&&map[s][i]!=0&&vis[i] == 0){

if(\_dis+map[s][i] > max){

max = \_dis+map[s][i];

}

maxdis(i,n,\_dis+map[s][i],max,vis);

}

}

}

void output(int n){

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

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

cout<<map[i][j]<<" ";

cout<<endl;

}

}

int main(){

int T;

cin>>T;

while(T--){

memset(map,0,sizeof(map));

memset(cnt,0,sizeof(cnt));

memset(dis,0,sizeof(dis));

memset(path,0,sizeof(path));

int n;

cin>>n;

for(int k = 1;k < n;k++){

int i,j,d;

cin>>i>>j>>d;

map[i][j] = d;

map[j][i] = d;

cnt[i]++;

cnt[j]++;

}

output(n);

int max = 0,max\_p,max\_e;

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

if(cnt[i] == 1)

{

memset(vis,0,sizeof(vis));

dfs(i,n,vis,max,max\_p);

break;

}

}

cout<<"最长路径起点:"<<max\_p<<endl;//找到距离任选的起点的最远点

memset(vis,0,sizeof(vis));

memset(dis,0,sizeof(dis));

memset(parent,0,sizeof(parent));

parent[max\_p] = max\_p;

max = 0;

dfs(max\_p,n,vis,max,max\_e);

cout<<"最长路径终点:"<<max\_e<<endl;

int n1 = parent[max\_e];

vector<int> v;

while(n1!=parent[n1]){

path[n1] = 1;

v.push\_back(n1);

n1 = parent[n1];

}//标记路径上的点并加入vector

path[max\_e] = 1;

path[max\_p] = 1;

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

cout<<parent[i]<<" ";

cout<<endl;

//---求得最远距离

int maxres = 0;

for(int i = 0;i < v.size();i++){

int sp = v[i];

memset(vis,0,sizeof(vis));

int \_maxd = 0;

maxdis(sp,n,0,\_maxd,vis);

if(\_maxd>maxres)

maxres=\_maxd;

}

cout<<maxres<<endl;

}

}

**贪吃蛇：**

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

int F(int n)

{

if(n==0) return 0;

else if(n==1) return 1;

else {

return F(n-1)+F(n-2);

}

}

int go[][2]={

{-1,0},

{0,1},

{1,0},

{0,-1}

};

int main()

{

int n,m,t;

while(scanf("%d%d%d",&n,&m,&t)!=EOF)

{

char a[101][101];

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

scanf("%s",a[i]);

int ans=0,nx=0,ny=0,cnt=0;

for(int i=0;i<=t;i++)

{

//printf("%d %d %d\n",i,nx,ny);

if(a[nx][ny]!='.')

{

int tmp=a[nx][ny]-'0';

ans+=tmp;

//printf("%c ",a[nx][ny]);

}

int x=nx+go[F(cnt)%4][0],y=ny+go[F(cnt)%4][1];

while(x<0||x>=n||y<0||y>=m)

{

cnt++;

x=nx+go[F(cnt)%4][0];

y=ny+go[F(cnt)%4][1];

}

nx=x;

ny=y;

// printf("%d %d %d\n",cnt,nx,ny);

}

printf("%d\n",ans);

}

return 0;

}

**仰望星空**

#include <iostream>

#include <stdio.h>

using namespace std;

char maze[101][101];

bool mark[101][101];

int n,m;

int go[8][2]={

{1,0},

{-1,0},

{1,1},

{1,-1},

{0,1},

{0,-1},

{-1,-1},

{-1,1}

};

void DFS(int x,int y)

{

for(int i=0;i<8;i++)

{

int nx=x+go[i][0];

int ny=y+go[i][1];

if(nx<1||nx>n||ny<1||ny>m)

continue;

if(maze[nx][ny]=='.')

continue;

if(mark[nx][ny]==true)

continue;

mark[nx][ny]=true;

DFS(nx,ny);

}

return ;

}

int main()

{

while(scanf("%d%d",&m,&n)!=EOF)

{

if(n==0&&m==0)

break;

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

scanf("%s",maze[i]+1);

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

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

{

mark[i][j]=false;

}

int ans=0;

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

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

{

if(maze[i][j]=='.')

continue;

if(mark[i][j]==true)

continue;

DFS(i,j);

ans++;

}

printf("%d\n",ans);

}

return 0;

}

**圆桌会议：**

#include <iostream>

#include <stdio.h>

using namespace std;

bool mark[100001];

int prime[100001];

int primeSize;

void init()

{

primeSize=0;

for(int i=2;i<=100000;i++)

{

if(mark[i]==true) continue;

prime[primeSize++]=i;

if(i>=1000)

continue;

for(int j=i\*i;j<=100000;j+=i)

mark[j]=true;

}

}

int main()

{

init();

int T;

while(scanf("%d",&T)!=EOF)

{

while(T--)

{

int n;

scanf("%d",&n);

int a[100000],flag=0;

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

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

for(int i=1;i<=n/2;i++)

{

if(n%i==0)

{

flag=1;

int cnt=1;

for(int j=0;j<i;j++)

{

int z=j+i;

do{

if(a[z]!=a[j])

{

flag=0;

//printf("%d",z);

}

cnt++;

z+=i;

}while(z<n);

if(flag&&cnt>=3)

{

puts("YES");

break;

}

else flag=0;

}

if(flag)

break;

}

}

if(!flag)

puts("NO");

}

}

// cout << "Hello world!" << endl;

return 0;

}