```
#include <iostream>
#include <string.h>
using namespace std;

long long b,res;
int main(){
	int T,t;
	cin>>T;
	string a;
	t=T;
	while(T--){
		res=0;
		cin>>a>>b;
		if(b<0)b=-b;
		int n=a.length();
		for(int i=0;i<n;i++){
			if(a[i]=='-')continue;
			res=(res*10+a[i]-'0')%b;
		}
		cout<<"Case "<<t-T<<": ";
		if(res==0)
			cout<<"divisible"<<endl;
		else
			cout<<"not divisible"<<endl;
	}
	return 0;
}
```

```
#include<iostream>
#include<string.h>
using namespace std;
int a[2005];
int b[2005];
int main()
{
	int n,i,j;
	while(cin>>n)
	{

		memset(a,0,sizeof(a));
		memset(b,0,sizeof(b));
		for(i=1;i<=n;i++)
		{
			cin>>a[i];
			b[a[i]]++;
		}
		for(i=2000;i>=1;i--)
		{
			b[i]=b[i]+b[i+1];
		}
		for(i=1;i<=n;i++)
		{
			if(i==1) cout<<b[a[i]+1]+1;
			else cout<<" "<<b[a[i]+1]+1;
		}
		cout<<endl;
	}
	return 0;
}
```

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
	long long n,m,c;
	while(cin>>n>>m)
	{
		c=0;
		while(m!=0)
		{
			c=c+n/m;
			n=n%m;
			swap(n,m);
		}
		cout<<c<<endl;
	}
	return 0;
}
```

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

double A,B,C,D,E,F,Ai,Bi,tan,sum,kefan;
int main() {
        cin>>A>>B>>C>>D>>E>>F;
        Ai=A*E,Bi=B*E;
        double max1=-1,max2=1;
        for(int a = 0;a*A*100<=F;a++)
    for(int b = 0;b*B*100<=F-a*A*100;b++)
    for(int c = 0;c*C<=F-a*A*100-b*B*100;c++)
    for(int d = 0;d*D<=F-a*A*100-b*B*100-c*C;d++)
    {
        tan = c*C+d*D;
        sum = A*a*100+B*b*100+tan;
        kefan = Ai*a+Bi*b;
        if(kefan<tan||sum>F)continue;

        if(sum!=0&&tan/sum>max2/max1){
            max1 = sum;
            max2 = tan;
        }
    }
    cout<<(int)max1<<" "<<(int)max2<<endl;
        return 0;
}
```

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

char A[10005];
char B[10005];

int fa, fb;

void swap(char *a, int i, int j)
{
    /**
     *  '1'的 asci 码是49, '0'的asci码是48, '1' - '0'之后是1， 而1本身的asci码就是1
     *  所以 数字字符减'0'可以实现字符转数字.
     *
     *  char是字符型，可是字符在计算机内部同样是用二进制数表示的，因此，从这个意义上讲字符也好，数字也好，一律是用二进制数表示的。
     *
     *  char是8位二进制，int是16位，只是范围有差别。
     *  所以char存数字没有问题
     *
     */
    char t = a[i]-'0';
    a[i] = a[j]-'0';
    a[j] = t;
}

void add(char *a, char *b)
{
    int la = strlen(a),lb = strlen(b);

    int i,j,c=0,s,l;

    //反转输入数字，使他符合习惯,使数字右对齐
    for(i=fa,j=la-1; i<=j; ++i,--j) swap(a,i,j);
    for(i=fb,j=lb-1; i<=j; ++i,--j) swap(b,i,j);

    for(i=fa; i<la||i<lb; ++i)
    {
        s = a[i] + b[i] + c;
        c = s/10;
        a[i] = s%10;
    }

    //存储结果的时候再次反转数组，方便计算
    a[i] = c;

    l = c ? i : i-1;

    if(fa) printf("-");

    for(i=l; i>=fa; --i) printf("%d", a[i]);
}

int cmp(char *a, char *b)
{
  //除去符号，比较两个数大小
    int i,j,la,lb;

    la = strlen(a);
    lb = strlen(b);

    if(la-fa>lb-fb)
      return 1;
    else if(la-fa<lb-fb)
      return 0;
    else
    {
      for(i=0; i<la&&a[i+fa]==b[i+fb]; ++i);
        return a[i+fa]>b[i+fb];
    }
}

void minus(char *a, char *b)
{
    char *t;
    int i,j,ft,la,lb,c,l,s;

    if(!cmp(a,b))
    {
        t=a;
        a = b;
        b = t;

        ft = fa;
        fa = fb;
        fb = ft;
    }

    la = strlen(a);
    lb = strlen(b);

    for(i=fa,j=la-1; i<=j; ++i,--j) swap(a,i,j);
    for(i=fb,j=lb-1; i<=j; ++i,--j) swap(b,i,j);

    c = 0;
    l = -1;

    for(i=0; i+fa<la; ++i)
    {
        s = a[i+fa]-b[i+fb]-c>=0 ? 0 : 1;
        // 如果a[i+fa] > b[i+fb]那么 (10+a[i+fa]-b[i+fb])%10对结果不影响，都是取个位
        // 如果a[i+fa] < b[i+fb]那么 10+a[i+fa]-b[i+fb]相当于借了一位然后相减，求模后也不影响结果
        // 写的漂亮
        a[i+fa] = (10+a[i+fa]-b[i+fb]-c)%10;
        l = a[i+fa] ? i+fa : l;
        c = s;
    }

    if(l<0)
        printf("0");
    else
    {
        if(fa) printf("-");
        for(i=l; i>=fa; --i) printf("%d", a[i]);
    }
}

int main()
{
    scanf("%s%s", A, B);
    /**
     *  如果 '-' == A[0] 说明A是负数，同时fa等于1,然后可以用fa表示数组的第一个数字的位置
     *  如果有'-' 则从 1开始，否则从0开始，写的很巧妙
     */

    fa = ('-'==A[0]);
    fb = ('-'==B[0]);

    if(fa^fb) // ^ 异或 同真为假 同假为假， 符号相异为真
        minus(A,B);
    else
        add(A,B);
}
```

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

int main()
{
        ll n,t;
        cin>>n;
        t=0;
                t += n / 2;
                t += n / 3;
                t += n / 5;
                t += n / 7;
                t -= n / 2 / 3;
                t -= n / 2 / 5;
                t -= n / 2 / 7;
                t -= n / 3 / 5;
                t -= n / 3 / 7;
                t -= n / 5 / 7;
                t += n / 2 / 3 / 5;
                t += n / 2 / 3 / 7;
                t += n / 2 / 5 / 7;
                t += n / 3 / 5 / 7;
                t -= n / 2 / 3 / 5 / 7;
                cout<<n-t<<endl;
        return 0;
}
```

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

int t;
int a[205][205];
int n,m,k;
int dp[40005];
int main(){
        scanf("%d",&t);
        while(t--){
                scanf("%d%d%d",&n,&m,&k);
                memset(a,0,sizeof(a));
                memset(dp,0,sizeof(dp));
                for(int i=1;i<=n;i++){
                        for(int j=1;j<=m;j++){
                                scanf("%d",&a[i][j]);
                                a[i][j]+=a[i][j-1];
                        }
                }
                for(int i=1;i<=n;i++){                  //从第一行到最后一行
                        for(int v=k;v>=0;v--){          //01背包
                                for(int j=1;j<=m;j++){     //表示前j瓶
                                        if(v-j>=0){
                                                dp[v]=max(dp[v],dp[v-j]+a[i][j]);
                                        }
                                }
                        }
                }
                printf("%d\n",dp[k]);
        }
}
```

```
#include<stdio.h>
#include<string.h>
int pre[1005];
int total;
int find(int r)//查找根节点
{
    return pre[r]==r?r:pre[r]=find(pre[r]);
}

void unite(int a,int b)
{
    int fa=find(a);
    int fb=find(b);
    if(fa!=fb)
    {
        pre[fa]=fb;total--;
    }
}

int main()
{
    int p1,p2,f1,f2;
    int n,m,k;
    scanf("%d",&n);
    while(n--)
    {
        //printf(":::::%d\n",n);
        memset(pre,0,sizeof(pre));
        scanf("%d %d",&m,&k);
        for(int i=1;i<=m;i++)pre[i]=i;
        total=m;
        //printf("::::::%d\n",total);
        for(int i=1;i<=k;i++)
        {
            scanf("%d %d",&p1,&p2);
            unite(p1,p2);
        }
        printf("%d\n",total);

    }
    return 0;
}
```

```
#include<iostream>
#include<string.h>
#include<queue>
using namespace std;

int vis[100005];
int n,m,sum;
struct str
{
	int x,sum;
};
int bfs(int a)
{
	memset(vis,0,sizeof(vis));
	queue<str>Q;
	str tmp,now;
	tmp.x=a;tmp.sum=0;
	Q.push(tmp);
	while(!Q.empty())
	{
		now=Q.front();
		Q.pop();
		if(now.x==m) return now.sum;
		if(now.x-1>=0 && now.x<=100000 && vis[now.x-1]==0)
		{
			vis[now.x-1]=1;
			tmp.sum=now.sum+1;
			tmp.x=now.x-1;
			Q.push(tmp);
		}
		if(now.x+1<=100000 && now.x>=0 && vis[now.x+1]==0)
		{
			vis[now.x+1]=1;
			tmp.sum=now.sum+1;
			tmp.x=now.x+1;
			Q.push(tmp);
		}
		if(now.x*2<=100000 &&now.x>=0 && vis[now.x*2]==0)
		{
			vis[now.x*2]=1;
			tmp.sum=now.sum+1;
			tmp.x=now.x*2;
			Q.push(tmp);
		}
	}
}
int main()
{
	while(cin>>n>>m)
	{
		cout<<bfs(n)<<endl;
	}
	return 0;
}
```