```c
//                  3n+1
#include<stdio.h>
int main(){
    int i,j,n;
    while(scanf("%d %d",&i,&j) != EOF){
        int max=0;
        printf("%d %d ",i,j);
        if(i>j){
            int temp = i;
            i = j;
            j = temp;
        }
        while(i<=j){
            n = i;
            int ans=1;
            while(n!=1){
                n = n%2==0 ? n/2 : 3*n+1;
                ans++;
            }
            i++;

            max = ans>max? ans:max;
        }
        printf("%d\n",max);
    }
    return 0;
}

```

```c
//                  UGLY NUMBER
#include<stdio.h>
int main(){
    int i=1,j=1,k=1,count=1;
    int arr[1500]={1};
    while(count<1500){
        int t1=arr[i-1]*2,t2=arr[j-1]*3,t3=arr[k-1]*5;

        int temp = (t1<=t2 && t1<=t3) ? t1: (t2<=t1 && t2<=t3)? t2:t3;
        if(temp == arr[count-1]){
            arr[count-1]==t1 ? i++: arr[count-1]==t2 ? j++:k++;
        }
        else{
        arr[count] = temp;
        arr[count]==t1 ? i++: arr[count]==t2 ? j++:k++;
        count++;
        }

    }
    printf("The 1500'th ugly number is %d.\n",arr[1500-1]);
    return 0;
}

```

```c
//                  kindergarden_counting_game
#include<stdio.h>

int main(){
    char str[1001];
    while(gets(str)){
        int i=0,count=0,check=0;
        while(str[i]!='\0'){
            if(str[i]>=65 &&str[i]<=90 || str[i]>=97 && str[i]<=122){
                check=1;
            }
            else{
                if(check){
                    count++;
                    check=0;
                }
            }
            i++;
        }
        if(check) count++;
        printf("%d\n",count);
    }
    return 0;
}

```

```c
//                  JOLLY JUMPERS
#include<stdio.h>
#include<math.h>
int main(){
    int n,arr[3000],dif[3000],i,j,jolly=1,found=1;
    while(scanf("%d",&n)!=EOF){
        jolly=1;
        for(i=0;i<n;i++){
            scanf("%d",&arr[i]);
        }
        for(i=0;i<n-1;i++){
            dif[i] = (int)(abs(arr[i+1]-arr[i]));
        }
        for(i=0;i<n-1;i++){
            found = 0;
            for(j=0;j<n-1;j++){
                if(dif[j]==i+1){
                    found = 1;
                    break;
                }
            }
            if(found==0){
                jolly=0;
                break;
            }
        }
        if(jolly) printf("Jolly\n");
        else printf("Not jolly\n");
    }
    return 0;
}

```

```c
//                  CONTEST A
#include<stdio.h>
int main(){
    char str[1001];
    while(gets(str)){
        int i=0,count=0,check=0;
        while(str[i]!='\0'){
            if(str[i]>=65 &&str[i]<=90 || str[i]>=97 && str[i]<=122){
                check=1;
            }
            else{
                if(check){
                    count++;
                    check=0;
                }
            }
            i++;
        }
        if(check) count++;
        printf("%d\n",count);
    }
    return 0;
}

```

```c
//                  CONTEST B
#include<stdio.h>
int main()
{
    int i,j,n;
    while(scanf("%d %d",&i,&j) != EOF)
    {
        int max=0;
        printf("%d %d ",i,j);

        int temp = i>=j ? i : j;
        i = i>=j ? j : i;
        j=temp;
        while(i<=j)
        {
            n = i;
            int ans=1;
            while(n!=1)
            {
                n = n%2==0 ? n/2 : 3*n+1;
                ans++;
            }
            i++;
            max = ans>max? ans:max;
        }
        printf("%d\n",max);
    }
    return 0;
}

```

```c
//                  CONTEST E
#include<stdio.h>
int GCD(int a,int b)
{
    int gcd=1;
    for(int i=1; i <= a && i<=b;i++)
    {
        if(a%i==0&&b%i==0)
            gcd = i;
    }
    return gcd;
}

int main()
{

    while(1)
    {
        int n;
        scanf("%d",&n);
        if(n==0) break;
        else
        {
            long long int G=0;
            for(int i=1; i<n; i++)
            {
                for(int j=i+1; j<=n; j++)
                {
                    G+=GCD(i,j);
                }
            }

            printf("%lld\n",G);
        }
    }
    return 0;
}

```

```c
//                  CONTEST F
#include<stdio.h>
long long int reverseAndAdd(long long int n){
    long long int a=0,temp=n;
    while(n>0){
        a*=10;
        a+=n%10;
        n/=10;
    }
    return a+temp;
}
int isPalindrom(long long int n){
    long long int a=0,temp=n;
    while(n>0){
        a*=10;
        a+=n%10;
        n/=10;
    }
    return a==temp;
}

int main(){
    int t;
    scanf("%d",&t);
    while(t){
        t--;
        int a;
        scanf("%d",&a);
        long long int n;
        n=a;
        int count=0;
        while(!isPalindrom(n)){
            n = reverseAndAdd(n);
            count++;
        }
        printf("%d %lld\n",count,n);

    }
    return 0;
}


```

```c
//                  CONTEST I
#include<stdio.h>
int main()
{

    int t;
    scanf("%d",&t);
    int cas=1;
    while(cas<=t)
    {
        int a,b,c;
        scanf("%d%d%d",&a,&b,&c);
        int maxm = a>=b ? a>=c ? a:c : b>=c ? b:c;
        int mini = a<=b ? a<=c ? a:c : b<=c ? b:c;
        printf("Case %d: %d\n",cas,a+b+c-maxm-mini);
        cas++;
    }
    return 0;
}

```

```c
//                  CONTEST K
#include<stdio.h>
int main(){

    int n;
    scanf("%d",&n);
    long long int a=1;
    for(int i=0;i<n;i++){
        a*=2;
    }
    printf("%lld\n",2*(a-1));
    return 0;
}

```

```c
//                  CONTEST L
#include<stdio.h>
int main(){
    int n,a;
    while(1){
        scanf("%d",&n);
        if(n==0) break;
        int row[n],col[n];
        for(int i=0;i<n;i++){row[i]=0;col[i]=0;}
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                scanf("%d",&a);
                col[j]+=a;
                row[i]+=a;
            }
        }
        int c1=0,c2=0;
        int r,c;
        for(int i=0;i<n;i++){
            if(row[i]%2==1){
                c1++;
                r=i+1;
            }
            if(col[i]%2==1){
                c2++;
                c=i+1;
            }
        }

        if(!(c1+c2)){
            printf("OK\n");
        }
        else if(c1==1 && c2==1){
            printf("Change bit (%d,%d)\n",r,c);
        }
        else{
            printf("Corrupt\n");
        }

    }
    return 0;
}

```