技巧

优读

```
#include<iostream>
#include<cstring>
#include<cstdio>
using namespace std;
int n,a[1010];
int inline read()
{
        int num=0;
        char c;
        bool plus=true;
        while((c=getchar())==' '||c=='\n'||c=='\r');
        if(c=='-') plus=false;
        else num=c-'0';
        while(isdigit(c=getchar()))
                num=num*10+c-'0';
        return  num*(plus?1:-1);
}
int main()
{
        n=read();
        for(int i=1;i<=n;i++)
                a[i]=read();
        printf("%d\n",n);
        for(int i=1;i<=n;i++)
                printf("%d ",a[i]);
        printf("\n");
        return 0;
}
```

斐波那契通项公式

```
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;
long double n=0;
//int p;  printf("%lld ",(long long)((pow(((1+sqrt(5))/2),n)-pow(((1-sqrt(5))/2),n))/sqrt(5)));
//f[2*n]=f[2*n-1]+...+f[3]+f[1];
//f[2*n+1]-1=f[2*n]+...+f[4]+f[2];
//f[2*n]=f[n]*f[n]+2*f[n]*f[n-1];
int main(){
        //scanf("%d",&p);
        for(int i=1;i<=92;i++){
                n++;
                printf("%lld ",(long long)((pow(((1+sqrt(5))/2),n)-pow(((1-sqrt(5))
                if(i%5==0) cout<<endl;
```

```
        }
        return 0;
}
```

斐波那契公约数

```
//luogu1306
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
long long n,m,c,d,p=100000000;
struct jz{
    long long a[40][40];
};
long long gcd(long long a,long long b){
    return b==0?a:gcd(b,a%b);
}
jz s;
jz fz(jz x){
    memset(x.a,0,sizeof(x.a));
    for(long long i=1;i<=n;i++){
        x.a[i][i]=1;
    }
    return x;
}
jz cf(jz x,jz y){
    jz neww;
    memset(neww.a,0,sizeof(neww.a));
    for(long long i=1;i<=n;i++){
        for(long long j=1;j<=n;j++){
            for(long long k=1;k<=n;k++){
                neww.a[i][j]=((x.a[i][k]*y.a[k][j])%p+neww.a[i][j])%p;
            }
        }
    }
    return neww;
}
jz speed(jz x,long long b){
    if(b==1) return x;
    jz cur,ans;
    ans=fz(ans);
    cur=x;
    while(b){
        if(b&1){
            ans=cf(ans,cur);
        }
        cur=cf(cur,cur);

        b>>=1;
    }
    return ans;
```

```
}
int main(){

    s.a[1][1]=1;
    s.a[1][2]=1;
    s.a[2][1]=1;
    s.a[2][2]=0;
    n=2;
    //scanf("%d",&m);
    scanf("%lld%lld",&c,&d);
    long long e=gcd(c,d);
    jz x=speed(s,e-1);
    printf("%lld\n",x.a[1][1]%p);
    return 0;
}
```

斐波那契矩阵快速幂

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int n,m;
struct jz{
        int a[40][40];
};
jz s;
jz fz(jz x){
        memset(x.a,0,sizeof(x.a));
        for(int i=1;i<=n;i++){
                x.a[i][i]=1;
        }
        return x;
}
jz cf(jz x,jz y){
        jz neww;
        memset(neww.a,0,sizeof(neww.a));
        for(int i=1;i<=n;i++){
                for(int j=1;j<=n;j++){
                        for(int k=1;k<=n;k++){
                                neww.a[i][j]+=x.a[i][k]*y.a[k][j];
                        }
                }
        }
        return neww;
}
jz speed(jz x,int b){
        if(b==1) return x;
        jz cur,ans;
        ans=fz(ans);
        cur=x;
        while(b){
                if(b&1){
```

```
                        ans=cf(ans,cur);
                }
                cur=cf(cur,cur);

                b>>=1;
        }
        return ans;
}
int main(){

        s.a[1][1]=1;
        s.a[1][2]=1;
        s.a[2][1]=1;
        s.a[2][2]=0;
        n=2;
        scanf("%d",&m);
        jz x=speed(s,m-1);
        printf("%d\n",x.a[1][1]);
        return 0;
}
```

接近分组

```
#include<bits/stdc++.h>
#define ll long long
#define db double
#define ld long double
using namespace std;
int n,m;
int main(){
        //cut n to m
        cin>>n>>m;
        for(int i=1;i<=m;i++){
                if(i!=1) printf(" ");
                cout<<ceil((n-i+1)/(db)m);
        }
        cout<<endl;
        return 0;
}
```