

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
struct matrix
```

```
{
```

```
    long long v[5][5];
```

```
    long long row,col;
```

```
};
```

```
long long mod;
```

```
matrix multi(matrix a, matrix b)
```

```
{
```

```
    assert(a.col == b.row);
```

```
    matrix r;
```

```
    r.row = a.row;
```

```
    r.col = b.col;
```

```
    for(long long i=0; i<r.row; i++)
```

```
        for(long long j=0; j<r.col; j++)
```

```
        {
```

```
            long long sum = 0;
```

```
            for(long long k=0; k<a.col; k++)
```

```
            {
```

```
                sum+=(a.v[i][k] * b.v[k][j]);
```

```
                sum%=mod;
```

```
            }
```

```
            r.v[i][j] = sum;
```

```
        }
```

```
    return r;
```

```
}
```

```
matrix power(matrix mat,long long p)
```

```
{
```

```
    assert(p >= 1 );
```

```
    if(p==1) return mat;
```

```
    if(p%2 == 1)
```

```
        return multi(mat,power(mat,p-1));
```

```
    matrix ret = power(mat,p/2);
```

```
    ret = multi(ret,ret);
```

```
    return ret;
```

```
}
```

```
int main()
```

```
{
```

```
    long long a,b,t,c,x,w = 0;
```

```
    cin>>t;
```

```
    while(t--)
```

```
    {
```

```
        cin>>x>>a>>b>>c;
```

```
        mod = 10007;
```

```
        matrix mt;
```

```
        mt.row = mt.col = 4;
```

```
        mt.v[0][0] =a, mt.v[0][1] = 0,mt.v[0][2] = b,mt.v[0][3] = c;
```

```
        mt.v[1][0] =1, mt.v[1][1] = 0,mt.v[1][2] = 0,mt.v[1][3] = 0;
```

```
        mt.v[2][0] =0, mt.v[2][1] = 1,mt.v[2][2] = 0,mt.v[2][3] = 0;
```

```
        mt.v[3][0] =0, mt.v[3][1] = 0,mt.v[3][2] = 0,mt.v[3][3] = 1;
```

```
        if(x<3)
```

```
        {
```

```

        printf("Case %lld: ",++w);
        if(x==0) cout<< 0<<endl;
        if(x==1) cout<< 0<<endl;
        if(x==2) cout<< 0<<endl;
    }
    else
    {

        long long aa = a*(a*c+c) + c;
        long long bb = a*c + c;
        long long cc = c;
        //cout << aa << " " << bb << " " << cc << endl;
        mt = power(mt,x-2);
        long long ans = mt.v[0][3];
        ans%=mod;
        printf("Case %lld: ",++w);
        cout<<ans<<endl;
    }

}

return 0;
}

```