题目大意：n个苹果往根上掉，第i个分支的苹果经过1s后会掉到第pi个分支，如果一个分支在某一个时刻有偶数个苹果，这些苹果全都会消失，否则保留下一个，当苹果掉到1号点就被视为捡到，问最多能捡到多少个苹果.

分析：这题不要想复杂了......

　　　把这棵树给构造出来，深度为i的苹果需要跳i次才能跳到根节点，它们必然会在根节点相遇. 那么对于每一个深度，数一下有多少苹果，如果有奇数个ans++就好了.

题意：给你一个树，每个节点都会长苹果，然后每一秒钟，苹果往下滚一个。两个两个会抵消苹果。问最后在根节点能收到多少个苹果。

解题思路：把树画出来，分析一下就会发现，只需要统计每一层有多少个苹果即可。因为每一层的苹果最后都会汇到根节点，所以直接判断每一层的苹果树是不是2的倍数即可。

#include<cstdio>

#include<iostream>

#include<vector>

using namespace std;

vector<int> child[200000];

int ceng[200000];

void dfs(int point,int layer)

{

ceng[layer]++;

for(int i=0;i<child[point].size();i++)

dfs(child[point][i],layer+1);

}

int main()

{

int n,x;

cin>>n;

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

{

cin>>x;

child[x].push\_back(i);

}

dfs(1,1);

int ans(0);

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

{

if(ceng[i]&1)ceng[i]=1;

else ceng[i]=0;

ans+=ceng[i];

}

cout<<ans<<"\n";

return 0;

}