题目大意：Farmer John一开始是站在树1下面，他可以在树1和树2之间来回移动W次，每分钟只能移动一次，也可以不移动每分钟有且只有一棵树会掉苹果，问在时间为T分钟内，他可以得到多少个苹果

分析：状态：dp[i][j]表示在第i分钟时，已经移动了j次后得到的苹果数量。

状态转移方程：dp[i][j] = max(dp[i-1][j], dp[i-1][j-1])，然后判断当前是否在第i分钟掉苹果的那颗树下，是的话，dp[i][j]++。

对状态转移方程的解释如下：第i分钟能得到的苹果数量，等于在第i-1分钟时，在树1和树2下得到苹果的最大值。j为偶数则在树1下面，奇数则在树2下面。

#include <iostream>

#include <iomanip>

#include <stdio.h>

#include <stdlib.h>

#include <algorithm>

#include <functional>

#include <vector>

#include <cmath>

#include <string>

#include <stack>

#include<cstring>

#include <queue>

using namespace std;

int dp[1010][100];

int a[1010];

int main()

{

// freopen("input.txt","r",stdin);

int w,t;

cin>>t>>w;

for(int i=0;i<t;i++)

cin>>a[i];

memset(dp,0,sizeof(dp));

if(a[0]==1)

dp[0][0]=1;

else

dp[0][1]=1;

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

for(int j=0;j<=w;j++)

{

if(j==0)//一次也没动

dp[i][j]=dp[i-1][j]+a[i]%2;//一次也没动就是在1的位置

else

{

dp[i][j]=max(dp[i-1][j],dp[i-1][j-1]);

if(a[i]==j%2+1)

dp[i][j]++;

}

}

int sum(0);

for(int i=0;i<=w;i++)//你不知道动了几次是最优的

sum=max(sum,dp[t-1][i]);

cout<<sum<<"\n";

return 0;

}