# 1176免费馅饼

**Problem Description**

都说天上不会掉馅饼，但有一天gameboy正走在回家的小径上，忽然天上掉下大把大把的馅饼。说来gameboy的人品实在是太好了，这馅饼别处都不掉，就掉落在他身旁的10米范围内。馅饼如果掉在了地上当然就不能吃了，所以gameboy马上卸下身上的背包去接。但由于小径两侧都不能站人，所以他只能在小径上接。由于gameboy平时老呆在房间里玩游戏，虽然在游戏中是个身手敏捷的高手，但在现实中运动神经特别迟钝，每秒种只有在移动不超过一米的范围内接住坠落的馅饼。现在给这条小径如图标上坐标：

IMG_256

为了使问题简化，假设在接下来的一段时间里，馅饼都掉落在0-10这11个位置。开始时gameboy站在5这个位置，因此在第一秒，他只能接到4,5,6这三个位置中其中一个位置上的馅饼。问gameboy最多可能接到多少个馅饼？（假设他的背包可以容纳无穷多个馅饼）

**Input**

输入数据有多组。每组数据的第一行为以正整数n(0<n<100000)，表示有n个馅饼掉在这条小径上。在结下来的n行中，每行有两个整数x,T(0<T<100000),表示在第T秒有一个馅饼掉在x点上。同一秒钟在同一点上可能掉下多个馅饼。n=0时输入结束。

**Output**

每一组输入数据对应一行输出。输出一个整数m，表示gameboy最多可能接到m个馅饼。  
提示：本题的输入数据量比较大，建议用scanf读入，用cin可能会超时。

**Sample Input**

6

5 1

4 1

6 1

7 2

7 2

8 3

0

**Sample Output**

4

代码清单：

#include<stdio.h>

#include<string.h>

#include<algorithm>

#include<iostream>

using namespace std;

int a[100005][12];

int main()

{

int n,m,num=0,t;

while(scanf("%d",&n)!=EOF,n)

{

memset(a,0,sizeof(a));

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

{

scanf("%d%d",&m,&t);

a[t][m]++;//第几分钟第几位置加上一个馅饼

if(t>num)//记录最大的行

num=t;

}

for(int i=num-1;i>=0;i--)//最后一行不用加，i从maxx行开始也可以，不影响结果

{

a[i][0]+=max(a[i+1][0],a[i+1][1]);//边界的数只能由上一行的两个数中取最大的

for(int j=1;j<10;j++)

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

a[i][10]+=max(a[i+1][10],a[i+1][9]);

}

printf("%d\n",a[0][5]);

}

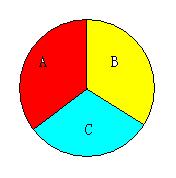
return 0;

}

# 2154跳舞毯

**Problem Description**

由于长期缺乏运动，小黑发现自己的身材臃肿了许多，于是他想健身，更准确地说是减肥。  
小黑买来一块圆形的毯子，把它们分成三等分，分别标上A,B,C，称之为“跳舞毯”，他的运动方式是每次都从A开始跳，每次都可以任意跳到其他块，但最后必须跳回A，且不能原地跳.为达到减肥效果，小黑每天都会坚持跳n次，有天他突然想知道当他跳n次时共几种跳法，结果想了好几天没想出来－\_－  
现在就请你帮帮他，算出总共有多少跳法。



**Input**

测试输入包含若干测试用例。每个测试用例占一行，表示n的值(1<=n<=1000)。  
当n为0时输入结束。

**Output**

每个测试用例的输出占一行，由于跳法非常多，输出其对10000取模的结果.

**Sample Input**

2

3

4

0

**Sample Output**

2

2

6

//简单dp

#include<iostream>

#include<cstring>

#include<cstdio>

#include<cmath>

using namespace std;

int dp[1001][3];

int main()

{

int n,i;

dp[1][2]=1;

dp[1][3]=1;dp[1][1]=0;

for(i=2;i<=1000;i++)

{

dp[i][1]=dp[i-1][2]+dp[i-1][3];

dp[i][2]=dp[i-1][1]+dp[i-1][3];

dp[i][3]=dp[i-1][1]+dp[i-1][2];

dp[i][1]=dp[i][1]%10000;

dp[i][2]=dp[i][2]%10000;

dp[i][3]=dp[i][3]%10000;

}

while(scanf("%d",&n)&&n)

{

printf("%d\n",dp[n][1]);

}

return 0;

}

# 2151 Worm

**Problem Description**

自从见识了平安夜苹果的涨价后，Lele就在他家门口水平种了一排苹果树，共有N棵。  
  
突然Lele发现在左起第P棵树上(从1开始计数)有一条毛毛虫。为了看到毛毛虫变蝴蝶的过程，Lele在苹果树旁观察了很久。虽然没有看到蝴蝶，但Lele发现了一个规律：每过1分钟，毛毛虫会随机从一棵树爬到相邻的一棵树上。  
  
比如刚开始毛毛虫在第2棵树上，过1分钟后，毛毛虫可能会在第1棵树上或者第3棵树上。如果刚开始时毛毛虫在第1棵树上，过1分钟以后，毛毛虫一定会在第2棵树上。  
  
现在告诉你苹果树的数目N，以及毛毛刚开始所在的位置P，请问，在M分钟后，毛毛虫到达第T棵树，一共有多少种行走方案数。

**Input**

本题目包含多组测试，请处理到文件结束(EOF)。  
每组测试占一行，包括四个正整数N,P,M,T(含义见题目描述，0<N,P,M,T<100)

**Output**

对于每组数据，在一行里输出一共的方案数。  
题目数据保证答案小于10^9

**Sample Input**

3 2 4 2

3 2 3 2

**Sample Output**

4

0

#include<iostream>

#include<string.h>

#include<cmath>

using namespace std;

int main()

{

int t,n,p,m,i,j;

int dp[101][101];

while(scanf("%d%d%d%d",&n,&p,&m,&t)!=EOF)

{

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

dp[0][p]=1;

for(i=1;i<=m;i++)

{

dp[i][1]=dp[i-1][2];

dp[i][n]=dp[i-1][n-1];

for(j=2;j<n;j++)

{

dp[i][j]=dp[i-1][j-1]+dp[i-1][j+1];

}

}

printf("%d\n",dp[m][t]);

}

return 0;

}