这道题麻烦的是概率这东西没法用个循环表示出来，根据我以往的经验，指望着把给出的测试数据乘上一百或者一万这种方法是完全不可取的。

乍一看，这道题目是拿概率当包，因为概率是浮点数，如果是想像普通背包一样处理，那是完全行不通的。

但是把题目转换一下，题目给出的概率是被抓的概率，我们可以把所有银行能抢的钱的总和当成一个包，把概率转换为安全概率，很容易理解这个包最后一定是能装满的，我们包里面的财富是什么？是概率！安全的概率！

从包的最大值往下查，找到第一个包里面装的安全概率可以让我们满意，这就是我们想要的结果。

#include <iostream>

#include<cstring>

#include<cstdio>

using namespace std;

int v[110];

double w[110];

double dp[10010];

int main()

{

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

int T,n;

double chance;

cin>>T;

while(T--)

{

cin>>chance>>n;

chance=1-chance;

int sum(0);

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

{

cin>>v[i]>>w[i];

sum+=v[i];

w[i]=1-w[i];//安全的概率

}

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

dp[0]=1;

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

for(int j=sum;j>=v[i];j--)

if(j>=v[i])

dp[j]=max(dp[j],dp[j-v[i]]\*w[i]);//选不选这个物品

for(int i=sum;i>=0;i--)

if(dp[i]>chance)

{

cout<<i<<"\n";

break;

}

}

return 0;

}