BM 板子

```
#include <bits/stdc++.h>
using namespace std;
#define sfi(a) scanf("%d",&a)
#define sfd(a) scanf("%lf",&a)
#define sfl(a) scanf("%lld",&a)
#define sfs(a) scanf("%s",a)
#define rep(i,a,b) for(int i=int(a);i<int(b);++i)
#define dwn(i,b,a) for(int i=int(b-1);i>=int(a);--i)
#define mem(a,p) memset(a,p,sizeof(a))
#pragma comment(linker,"/STACK:102400000,102400000")
typedef long long LL;
typedef unsigned UINT;
typedef unsigned long long ULL;
const int MAXN=10005;
struct BM
{
	int n;
	vector<double> ps[MAXN];
	int pn,fail[MAXN];
	double delta[MAXN];
	void Solve(double *x,int n)
	{
		pn=0;
		mem(fail,0);
		mem(delta,0);
		ps[0].clear();
		rep(i,1,n+1)
		{
			double dt=-x[i];
			rep(j,0,ps[pn].size())
				dt+=x[i-j-1]*ps[pn][j];
			delta[i]=dt;
			if(fabs(dt)<=1e-8)continue;
			fail[pn]=i;
			if(!pn)
			{
				ps[++pn].resize(1);
				continue;
			}
			vector<double> &ls=ps[pn-1];
			double k=-dt/delta[fail[pn-1]];
			vector<double> cur;
			cur.resize(i-fail[pn-1]-1);
			cur.push_back(-k);
			rep(j,0,ls.size())cur.push_back(ls[j]*k);
			if(cur.size()<ps[pn].size())cur.resize(ps[pn].size());
			rep(j,0,ps[pn].size())cur[j]+=ps[pn][j];
			ps[++pn]=cur;
		}
	}

	void print()
	{
		rep(g,0,ps[pn].size())
			printf("%lf ",ps[pn][g]);
		printf("\n");
	}
}B;
double x[MAXN];
int main()
{
	int n;
	while(~sfi(n))
	{
		rep(i,1,n+1)
		sfd(x[i]);
		B.Solve(x,n);
		B.print();
	}
}
```