```cpp
/*** Matrix Chain Multiplication:
#include<bits/bits/stdc++.h>
using namespace std;
#define ll long long
ll n,r[15],c[15],dp[15][15],a[15],g[15];
ll fun(ll b,ll e)
{
    if(b>=e)return 0;
    if(dp[b][e]!=-1)return dp[b][e];
    ll ret = 10000000000000000;
    for(ll m=b; m<e; m++){
        ll sum = fun(b,m)+fun(m+1,e)+(r[b]*c[m]*c[e]);
        ret = min(ret,sum);
    }
    return dp[b][e]=ret;
}
void path(ll b,ll e)
{
    if(b>=e)return;
    ll ret = fun(b,e);
    for(ll m=b; m<e; m++)
    {
        ll sum = fun(b,m)+fun(m+1,e)+(r[b]*c[m]*c[e]);
        if(sum==ret)
        {
            printf("("); if(b==m)printf("A%d",b);
            path(b,m);
            printf(" x ");
            path(m+1,e);
            if(m+1==e)printf("A%d",e); printf(")");
            break;
        }
    }
}
int main()
{
    int ks=0;
    while(scanf("%lld",&n) && n)
    {
        for(int i=1; i<=n; i++) scanf("%lld%lld",&r[i],&c[i]);

        if(n==1){ printf("Case %d: (A1)\n",++ks); continue; }

        memset(dp,-1,sizeof(dp));
        ll ans = fun(1,n);

        printf("Case %d: ",++ks);
        path(1,n);
        printf("\n");
    }
    return 0;
}
```