

```
#include<bits/stdc++.h>

using namespace std;

#define ll unsigned long long
```

```
struct matrix
```

```
{
    ll mat[4][4];

    ll row, col;
};
```

```
matrix multiply(matrix a, matrix b)
```

```
{
    assert(a.col == b.row);

    matrix r;

    r.row = a.row;
    r.col = b.col;

    for(int i=0; i<r.row; i++)
        for(int j=0; j<r.col; j++)
        {
            ll sum = 0;

            for(int k=0; k<a.col; k++)
                sum+=a.mat[i][k] * b.mat[k][j];

            r.mat[i][j] = sum;
        }

    return r;
}
```

```
matrix power(matrix a, ll p)
```

```
{
    assert(p>=1);

    if(p == 1) return a;
```

```

else if(p%2 == 1)

    return multiply(a,power(a,p-1));


matrix ret = power(a,p/2);
return multiply(ret,ret);
}


int main()
{
    ll a_b,ab,t,n,w = 0;

    scanf("%llu",&t);
    while(t--)
    {
        scanf("%llu %llu %llu",&a_b,&ab,&n);

        printf("Case %llu: ",++w);

        if(n == 0)

            printf("2\n");

        if(n == 1)

            printf("%llu\n",a_b);

        if(n == 2)

            printf("%llu\n",(a_b * a_b) - 2 * ab);
        else
        {
            matrix base;

            base.mat[0][0] = a_b;

            base.mat[0][1] = (-1)* ab;

            base.mat[1][0] = 1;

            base.mat[1][1] = 0;

            base.row = 2;

            base.col = 2;

            matrix ans = power(base,n-2);

```

```
    printf("%llu\n",a_b*ans.mat[0][1]+((a_b * a_b) - 2 * ab) *ans.mat[0][0]);  
}  
  
}  
return 0;  
}
```