```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int MAXN=100010;
struct node
{
    ll sum;
    ll lazy1;//乘法;
    ll lzay2;//加法;
    ll l,r;
}tree[MAXN<<2];
ll n,mod,m;


void change1(long long,long long,long long,long long);
void change2(long long,long long,long long,long long);
inline long long lread()
{
    long long res = 0; bool bo = 0; char c;
    while (((c = getchar()) < '0' || c > '9') && c != '-');
    if (c == '-') bo = 1; else res = c - 48;
    while ((c = getchar()) >= '0' && c <= '9')
        res = (res << 3) + (res << 1) + (c - 48);
    return bo ? ~res + 1 : res;
}
inline void lprint(long long x)
{
    if (x>=10)     lprint(x/10);
    putchar(x%10+'0');
}
void build(long long tr,long long l,long long r)
{
    tree[tr].l=l;
    tree[tr].r=r;
    tree[tr].lazy1=1;
    if(l==r)
    {
        tree[tr].sum=lread();
        return ;
    }
    ll mid=(l+r)>>1;
    build(tr<<1,l,mid);
    build(tr<<1|1,mid+1,r);
    tree[tr].sum=(tree[tr<<1].sum+tree[tr<<1|1].sum);
}
void change1(ll tr,ll l,ll r,ll d)
{
    if(tree[tr].l==l&&tree[tr].r==r)
    {
        tree[tr].sum=tree[tr].sum*d%mod;
        tree[tr].lazy1=tree[tr].lazy1*d%mod;
        tree[tr].lzay2=tree[tr].lzay2*d%mod;
        return ;
    }
    ll mid=(tree[tr].l+tree[tr].r)>>1;
    if(tree[tr].lazy1!=1)
    {
        change1(tr<<1,tree[tr].l,mid,tree[tr].lazy1);
        change1(tr<<1|1,mid+1,tree[tr].r,tree[tr].lazy1);
        tree[tr].lazy1=1;
    }
    if(tree[tr].lzay2!=0)
```

```
    {
        change2(tr<<1,tree[tr].l,mid,tree[tr].lzay2);
        change2(tr<<1|1,mid+1,tree[tr].r,tree[tr].lzay2);
        tree[tr].lzay2=0;
    }
    if(r<=mid) change1(tr<<1,l,r,d);
    else if(l>mid) change1(tr<<1|1,l,r,d);
    else
    {
        change1(tr<<1,l,mid,d);
        change1(tr<<1|1,mid+1,r,d);
    }
    tree[tr].sum=(tree[tr<<1].sum+tree[tr<<1|1].sum)%mod;
}
void change2(ll tr,ll l,ll r,ll d)
{
    if(tree[tr].l==l&&tree[tr].r==r)
    {
        tree[tr].sum=(tree[tr].sum+d*(r-l+1))%mod;
        tree[tr].lzay2=(tree[tr].lzay2+d)%mod;
        return;
    }
    ll mid=(tree[tr].l+tree[tr].r)>>1;
    if(tree[tr].lazy1!=1)
    {
        change1(tr<<1,tree[tr].l,mid,tree[tr].lazy1);
        change1(tr<<1|1,mid+1,tree[tr].r,tree[tr].lazy1);
        tree[tr].lazy1=1;
    }
    if(tree[tr].lzay2!=0)
    {
        change2(tr<<1,tree[tr].l,mid,tree[tr].lzay2);
        change2(tr<<1|1,mid+1,tree[tr].r,tree[tr].lzay2);
        tree[tr].lzay2=0;
    }
    if(r<=mid) change2(tr<<1,l,r,d);
    else if(l>mid) change2(tr<<1|1,l,r,d);
    else
    {
        change2(tr<<1,l,mid,d);
        change2(tr<<1|1,mid+1,r,d);
    }
    tree[tr].sum=(tree[tr<<1].sum+tree[tr<<1|1].sum)%mod;
}
long long query(ll tr,ll l,ll r)
{
    ll mid=(tree[tr].l+tree[tr].r)>>1;
    if(tree[tr].l==l&&tree[tr].r==r) return tree[tr].sum;//%mod
    if(tree[tr].lazy1!=1)
    {
        change1(tr<<1,tree[tr].l,mid,tree[tr].lazy1);
        change1(tr<<1|1,mid+1,tree[tr].r,tree[tr].lazy1);
        tree[tr].lazy1=1;
    }
    if(tree[tr].lzay2!=0)
    {
        change2(tr<<1,tree[tr].l,mid,tree[tr].lzay2);
        change2(tr<<1|1,mid+1,tree[tr].r,tree[tr].lzay2);
        tree[tr].lzay2=0;
    }
    if(r<=mid) return query(tr<<1,l,r);//%mod
    else if(l>mid) return query(tr<<1|1,l,r);//%mod
```

```
    else return (query(tr<<1,l,mid)+query(tr<<1|1,mid+1,r));//%%mod
}
int main()
{
//      freopen("testdata.in","r",stdin);
    n=lread();m=lread();mod=lread();
    build(1,1,n);
    while(m--)
    {
        ll g=lread(),l=lread(),r=lread();
        if(g==1)
        {
            ll d=lread();
            change1(1,l,r,d);
        }
        if(g==2)
        {
                        ll d=lread();
                        change2(1,l,r,d);
        }
        if(g==3)
        {
//            lprint(query(1,l,r)%mod);
            cout<<(query(1,l,r)%mod);
            putchar('\n');
        }
    }
    return 0;
}
```