A. A Math Challenge

题意

$$ \sum_{i=0}^n\sum_{1\le cj\le ai+b}i^pj^q $$

题解

设 $F(n)=\sum_{i=1}^n i^q$ 于是上式转化为

$$ \sum_{i=0}^ni^pF(\lfloor \frac {ai+b}c\rfloor) $$

由于 $F(n)$ 是 $q+1$ 次多项式，所以高斯消元可以暴力求出 $F(n)$ 的表达式。于是问题转化为计算 $\sum_{i=0}^ni^p(\lfloor \frac {ai+b}c\rfloor)^k(0\le k\le q)$

上式用万能欧几里得算法板子可以 $O\left(p^2q^2\log c\right)$ 题目正解是类欧几里得算法 $O\left((p+q)^3\log c\right)$ 不过卡卡常还是能过的。

```
const int mod=998244353,MAXK=53;
int C[MAXK][MAXK];
struct Node{
    int cntr,cntu,f[MAXK][MAXK];
    Node(int cntr=0,int cntu=0){
        this->cntr=cntr;
        this->cntu=cntu;
        mem(f,0);
    }
    Node operator * (const Node &b)const{
        static int px[MAXK],py[MAXK];
        static int b1[MAXK][MAXK],b2[MAXK][MAXK];
        Node c;
        int dx=cntr,dy=cntu;
        px[0]=py[0]=1;
        _for(i,1,MAXK)
        px[i]=1LL*px[i-1]*dx%mod;
        _for(i,1,MAXK)
        py[i]=1LL*py[i-1]*dy%mod;
        _for(i,0,MAXK)_rep(j,0,i){
            b1[i][j]=1LL*C[i][j]*px[i-j]%mod;
            b2[i][j]=1LL*C[i][j]*py[i-j]%mod;
        }
        c.cntr=(cntr+b.cntr)%mod;
        c.cntu=(cntu+b.cntu)%mod;
        _for(i,0,MAXK)_for(j,0,MAXK){
            c.f[i][j]=f[i][j];
            _rep(i2,0,i)_rep(j2,0,j)
c.f[i][j]=(c.f[i][j]+1LL*b.f[i2][j2]*b1[i][i2]%mod*b2[j][j2])%mod;
        }
```

```
        return c;
    }
};
Node quick_pow(Node n,int k){
    Node ans=Node(0,0);
    while(k){
        if(k&1)ans=ans*n;
        k>>=1;
        if(k)n=n*n;
    }
    return ans;
}
Node asgcd(int a,int b,int c,int n,Node su,Node sr){
    if(a>=c)
    return asgcd(a%c,b,c,n,su,quick_pow(su,a/c)*sr);
    int m=(1LL*a*n+b)/c;
    if(!m)
    return quick_pow(sr,n);
    else
    return quick_pow(sr,(c-b-1)/a)*su*asgcd(c,(c-
b-1)%a,a,m-1,sr,su)*quick_pow(sr,n-(1LL*c*m-b-1)/a);
}
Node cal(int a,int b,int c,int n){
    Node su=Node(0,1),sr=Node(1,0);
    _for(i,0,MAXK)
    sr.f[i][0]=1;
    return quick_pow(su,b/c)*asgcd(a,b%c,c,n,su,sr);
}
int a[MAXK][MAXK],pw[MAXK][MAXK],A[MAXK];
void Init(){
    C[0][0]=1;
    _for(i,1,MAXK){
        C[i][0]=1;
        _rep(j,1,i)
        C[i][j]=(C[i-1][j-1]+C[i-1][j])%mod;
    }
    _for(i,0,MAXK){
        pw[i][0]=1;
        _for(j,1,MAXK)
        pw[i][j]=1LL*pw[i][j-1]*i%mod;
    }
}
int quick_pow(int n,int k){
    int ans=1;
    while(k){
        if(k&1)ans=1LL*ans*n%mod;
        n=1LL*n*n%mod;
        k>>=1;
    }
    return ans;
}
```

```
void build(int n){
    _for(i,0,n){
        _for(j,0,n)
        a[i][j]=pw[i][j];
        _rep(j,1,i)
        a[i][n]=(a[i][n]+pw[j][n-2])%mod;
    }
    _for(i,0,n){
        int pos=-1;
        _for(j,i,n){
            if(a[j][i]){
                pos=j;
                break;
            }
        }
        if(pos!=i)
        swap(a[i],a[pos]);
        int div=quick_pow(a[i][i],mod-2);
        _rep(j,i,n)
        a[i][j]=1LL*a[i][j]*div%mod;
        _rep(j,0,n){
            if(j==i)continue;
            for(int k=n;k>=i;k--)
            a[j][k]=(a[j][k]+mod-1LL*a[j][i]*a[i][k]%mod)%mod;
        }
    }
    _for(i,0,n)
    A[i]=a[i][n];
}
int main()
{
    Init();
    int
a=read_int(),b=read_int(),c=read_int(),p=read_int(),q=read_int(),n=read_int
();
    Node ans=cal(a,b,c,n);
    int base=1;
    _for(i,0,MAXK){
        ans.f[0][i]=(ans.f[0][i]+base)%mod;
        base=1LL*base*(b/c)%mod;
    }
    build(q+2);
    int s=0;
    _rep(i,0,q+1)
    s=(s+1LL*ans.f[p][i]*A[i])%mod;
    enter(s);
    return 0;
}
```