struct np{

short n;

short p;

};

class big\_int{

private:

string num;

bool negative = false;

np cal[100]={{0}};

public:

int len = 0;

big\_int();

big\_int(string& c);

big\_int(const big\_int &b);

~big\_int();

big\_int operator+(big\_int q);

big\_int operator-(big\_int q);

big\_int operator-();

big\_int operator\*(big\_int q);

big\_int operator/(big\_int q);

big\_int operator/(int n);

big\_int operator%(big\_int q);

big\_int operator%(int n);

int operator[](int n);

bool operator>(big\_int q);

bool operator<(big\_int q);

bool operator==(big\_int q);

bool isPrime();

void operator=(big\_int q);

void operator=(string s);

big\_int abs();

big\_int square();

void check();

void print();

big\_int merge(big\_int q);

friend ostream& operator<<(ostream& out,const big\_int& q);

friend istream& operator>>(istream& in,const big\_int& q);

friend string to\_string(int n);

};

string to\_string(big\_int n);

ostream& operator<<(ostream& out,const big\_int& q);

istream& operator>>(istream& in,big\_int& q);

big\_int find\_object(string s,int nameCnt,big\_int\*\* ptr,string name[]);

big\_int calculate(string cs, big\_int\*\* ptr,string name[],int nameCnt);