Pràctiques Integrades
1er de Matemàtiques

Pràctica 15

curs 2002–03

15 Treballar amb nombres enters i congruències

Maple proporciona funcions per a realitzar les principals manipulacions de l'aritmètica amb nombres enters. Veurem a continuació algunes de les més significatives.

15.1 Nombres primers

Si us en recordeu, en la primera pràctica ja es presentava la funció `ifactor`, que determina la descomposició en factors primers d'un nombre enter. Com que el resultat que dóna aquesta funció no és fàcilment manipulable (en particular, no és fàcil extreure d'aquest resultat quins són els primers que divideixen al nombre que tenim, ni les potències d'aquests factors) existeix també la funció `ifactors` que dóna com a resultat una llista amb els factors primers, i les potències d'aquests factors, amb els que descompon un nombre enter donat.

Exercici 15.1

Considereu $n = 2^3 \cdot 5^4 \cdot 13^5$. Intenteu tornar a obtenir els factors 2, 5 i 13 i les potències respectives 3, 4 i 5 del resultat de la comanda `ifactor(n);`.

Utilitzeu ara `ifactors(n);` (i alguna cosa més) per a recuperar, separadament, els valors dels factors primers i de les potències corresponents de la descomposició del nombre n posant aquests resultats en dues llistes (una que sigui `f` per als factors i l'altra que sigui `p` per a les potències).

També és possible verificar si un nombre donat és primer o no sense necessitat de fer la seva descomposició. La funció que s'encarrega d'això és `isprime`. Podeu veure en l'exemple següent com funciona.

Exemple 15.1

```
> isprime(4);
> isprime(7);
> provap:= n-> if isprime(n) then print("és primer")
> else print("no és primer") end if;
> provap(456);
```

Noteu que la funció dóna com a resultat un valor cert o fals que es pot utilitzar com a condicional d'una comanda `if`.

Altres comandes relacionades amb els nombres primers són `ithprime(i)` (que dóna el primer que ocupa el lloc `i` en la llista dels primers ordenats), `nextprime(i)` (que dóna com a resultat el nombre primer més petit dels que són més grans que el nombre `i`) i `prevprime(i)` (que dóna el primer que és més gran entre els que són menors que `i`). Practiqueu amb alguns valors de `i` i mireu què passa quan `i` és molt gran.

15.2 Divisió entera

Cada cop que realitzem una divisió amb enters, són igualment significatius el quocient i la resta d'aquesta divisió. Les comandes de Maple `iquo(m,n)` i `irem(m,n)` donen com a resultat nombres q i r tals que $m = q \cdot n + r$ (amb $|r| < |n|$ i m i r amb el mateix signe).

Exemple 15.2

```
> iquo(158,18);
```

Les comandes `iquo` i `irem` tenen un tercer argument opcional que assigna a una variable qualsevol el valor del resultat de l'altra comanda. Així podem fer, per exemple,

Exemple 15.3

```
> iquo(487,18,'r');
> r;
> irem(487,18,'q');
> q;
```

(Les cometes de `'q'` i `'r'` serveixen per a poder fer l'assignació de valors sense preocuparse si les variables `q` i `r` ja tenien valors assignats).

Utilitzant aquestes instruccions podeu fer l'exercici següent:

Exercici 15.2

Feu un procediment que tingui com arguments un parell de nombres enters i que vagi donant els resultats parcials dels càlculs que es realitzen per a calcular el màxim comú divisor d'aquests dos nombres utilitzant l'algoritme d'Euclides (s'hauria de veure els diferents dividends i divisors i els quocients i restes de cada divisió, fins arribar a la divisió de resta 0).

Apliqueu la funció a uns quants parells de nombres triats a l'atzar.

Tot i que el procediment de l'exercici anterior calcula el màxim comú divisor de dos nombres, com ja podeu suposar ja hi ha comandes de Maple programades que calculen el màxim comú divisor i el mínim comú múltiple de dos nombres enters. La comanda `igcd(m1, m2, m3,...)` dóna com a resultat el màxim comú

divisor dels nombres enters `m1, m2, m3,...`, mentre que la comanda `ilcm(m1, m2, m3, ...)` calcula el mínim comú múltiple.

Exemple 15.4

Proveu les comandes anteriors amb

```
> igcd(868,784,900);
> ilcm(25,350,875);
```

i amb un parell d'exemples triats per vosaltres mateixos.

Existeix també una altra comanda que calcula el màxim comú divisor de dos nombres enters, encara més útil que `igcd`. La comanda `igcdex(m,n)` dóna igualment el màxim comú divisor de `m` i `n` però té un parell d'arguments opcionals que permeten conèixer un parell de nombres `a, b` tals que

$$\texttt{igcd(m,n)} = \texttt{a} \cdot \texttt{m} + \texttt{b} \cdot \texttt{n}$$

En l'exemple següent podeu veure un exemple de com fer servir aquesta comanda

Exemple 15.5

```
> igcdex(258,438);
> igcdex(258,438,'a','b');
> a;
> b;
> a*258+b*438;
```

Noteu que s'ha fet servir la mateixa estratgia de les comentes `' '` que en les comandes `iquo` i `irem` per assignar a les variables `a` i `b` el valors que ens interessava guardar.

15.3 Reducció mòdul p

Maple també té funcions per a treballar *mòdul p*. El resultat de l'expressió `m mod p` o de la funció `modp(m,p)` és el residu entre 0 i $p-1$ de m mòdul p.

Exemple 15.6

Podeu provar:

```
> 378946 mod 7;
> 345*267 mod 4;
> 985*x mod 7;
> modp(378946, 7);
> modp(345*267, 4);
> modp(985*x, 7);
```

Amb `mod` i `modp` també podeu mirar si dos nombres coincideixen, o no, mòdul un p donat.

Exemple 15.7

```
> evalb(378946 mod 7= 1 mod 7);
> evalb(345*267 mod 4=2 mod 4);
```

Si en lloc d'utilitzar `modp(m,p)` utilitzeu `mods(m,p)` (m'dul amb signe) obtindreu un resultat entre $-\frac{p}{2}$ i $\frac{p}{2}$.

15.4 Solucions enteres d'equacions

Existeix també una comanda específica per a determinar *totes* les solucions enteres d'una equació. Aquesta comanda és `isolve` i s'utilitza gairebé de la mateixa manera que `solve`. És a dir, la sintaxi normal serà `isolve({eq1, eq2, ...})` on `eq1, eq2, ...` són les equacions de les que es volen trobar les solucions enteres comuns.

Exemple 15.8

Les solucions enteres del sistema d'equacions $\left.\begin{array}{r} x^3+a\cdot x=14 \\ a^2-x=7 \end{array}\right\}$ es poden determinar amb

```
> isolve({x^3+a*x=14, a^2-x=7});
```

Exercici 15.3

Quines són les solucions enteres de l'equació dels triangles rectangles $x^2+y^2=z^2$?

Si heu vist la manera en que es mostren les solucions de l'equació dels triangles pitagòrics de l'exercici anterior, veureu que és molt útil la possibilitat d'afegir un argument opcional a la comanda `isolve`, que permet assignar un nom concret als diferents paràmetres que poden sortir en el resultat. En l'exemple següent podeu veure com fer una construcció d'aquest estil

Exemple 15.9

```
> isolve(x^2+y^2=5*z^2,{a,b,c});
```

Com podreu veure, els paràmetres dels que depèn la solució seran `a, b, c`.

Quina és la solució de l'equació anterior si preneu $a=1$, $b=-1$ i $c=3$?

Tingueu en compte que el problema de determinar les solucions enteres d'una equació arbitrària és tremendament complicat i que Maple *només fa el que sap i pot*. Per exemple, Maple no queda gaire bé

intentant resoldre l'equació $x^3 - 3y^2 = 0$ (que, com a mínim es pot solucionar amb $x = 0$, $y = 0$ i amb $x = y = 3$).

Exemple 15.10

```
> isolve(x^3-3*y^2=0);
```

15.5 Solucions de sistemes de congruències

Un problema bastant típic quan es treballa amb enters, principalment quan es plantegen qüestions relatives a reparticions, consisteix en resoldre un sistema de congruències de la forma

$$x \equiv a \pmod{\alpha}$$
$$x \equiv b \pmod{\beta}$$

(o potser amb més equacions).

Quan α i β no tenen factors comuns (el seu màxim comú divisor és 1) la comanda de Maple `chrem` pot obtenir directament les solucions. Si es vol resoldre el sistema anterior, la sintaxi que s'ha d'utilitzar és `chrem([a, b], [alpha, beta])` i el resultat serà l'únic nombre enter entre 0 i $\alpha \cdot \beta - 1$ que compleix les dues equacions (totes les altres solucions es poden obtenir sumant múltiples de $\alpha \cdot \beta$ a aquesta solució).

Exemple 15.11

```
> s:=chrem([1,3],[5,9]);
> s mod 5; s mod 9;
> s+45 mod 5; s+45 mod 9;
```

Quan els nombres α i β no són coprimers la comanda `chrem` no funciona i, per tant, no ens permet resoldre directament el problema. Això és només un petit inconvenient, ja que Maple posa a la nostra disposició totes les eines necessàries per a poder determinar la solució de qualsevol sistema de congruències del tipus que hem plantejat al principi d'aquest apartat.

Probablement ja sabeu

a) que el sistema $x \equiv a \pmod{\alpha}$, $x \equiv b \pmod{\beta}$ té solució si, i només si, $a - b$ és un múltiple del màxim comú divisor de α i β,

b) que la diferència entre dues solucions sempre és un múltiple del mínim comú múltiple de α i β,

c) que una solució particular es pot determinar prenent k i l tals que $k \cdot \alpha - l \cdot \beta = b - a$ i considerant $x = a + k \cdot \alpha = b + l \cdot \beta$.

Exercici 15.4

Feu un procediment que utilitzi el mateix conveni que `chrem` per a interpretar els arguments (dues llistes, una dels valors a, b i l'altre dels mòduls de les equacions α, β), que determini si el sistema és compatible i que,

quan ho sigui, doni com a resultat una seqüència amb una solució particular x_0 i el mínim comú múltiple de α i β per a un sistema de la forma

$$x \equiv a \pmod{\alpha}$$
$$x \equiv b \pmod{\beta}$$

Comproveu que el vostre procediment funciona per a l'equació

$$x \equiv 2 \pmod{6}$$
$$x \equiv 4 \pmod{8}$$

No hauria de ser ara massa complicat fer el següent

Exercici 15.5

Modifiqueu el procediment anterior per a poder resoldre un sistema de congruències amb un nombre arbitrari d'equacions. Fixeu-vos que l'únic que cal fer és anar reduint el nombre d'equacions substituint les dues primeres per la seva solució (que torna a venir donada per una equació del mateix tipus).