32 En aquest exercici treballarem amb l'estructura `quaternio`. Aquesta estructura contindrà un escalar de nom `real` i un vector de tres components de nom `imag`. Generalment es representa un quaternió de la forma

$$p = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k},$$

on $p_0, p_1, p_2, p_3 \in \mathbb{R}$ i $\mathbf{i}, \mathbf{j}$ i $\mathbf{k}$ són els quaternions unitaris fonamentals, i juntament amb 1 formen la base. Podem escriure també un quaternió p com $p = p_0 + \mathbf{u}$, amb $\mathbf{u} = (u_1, u_2, u_3) = u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$. Denotem per $\mathbb{H}$ el conjunt de quaternions amb valors en $\mathbb{R}$. Observem que $\mathbb{H}$ pot ser identificat amb $\mathbb{R}^4$ i el subespai de quaternions imaginaris purs amb $\mathbb{R}^3$.

Definirem les operacions fonamentals en $\mathbb{H}$ de la següent manera:

- La suma de dos quaternions $p = p_0 + \mathbf{u}, q = q_0 + \mathbf{v}$ és $p + q = (p_0 + q_0) + (\mathbf{u} + \mathbf{v})$.
- El producte d'un quaternió per un escalar: $\lambda \in \mathbb{R}$ i $p \in \mathbb{H}$, és $\lambda p = \lambda p_0 + \lambda \mathbf{u}$.
- El producte de quaternions ve definida pel producte dels elements de la base que és la següent:

  $$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1.$$

  La resta de productes es poden determinar a partir d'aquestes identitats.

  Atenció: el producte no és commutatiu.

a) Programeu les funcions `sumaQuat` i `prodQuat` corresponents a la suma i producte entre quaternions; reben dos quaternions i retornen un nou quaternió amb el resultat de l'operació. Escriviu també una funció `prodEscQuat` que calcula el producte d'un quaternió per un escalar; rep un quaternió i un escalar i retorna un nou quaternió. Aquestes funcions es guardaran en el fitxer **oper-quater.c**. Es crearà un fitxer **oper-quater.h** on es posaran la definició de l'estructura i les capçaleres corresponents a les funcions.

El conjunt de quaternions a valors reals, $\mathbb{H}$, amb les operacions suma i producte és un anell no commutatiu tal com el conjunt de matrius, és a dir, si $p, q \in \mathbb{H}$, aleshores pot ser $pq \neq qp$. Això fa que s'hagui d'anar en compte al fer el producte si és per l'esquerra o per la dreta.

b) Comproveu, fent servir les definicions i programant una funció `main` (que guardareu en **comprova.c**) que cridi les vostres funcions, que donats dos quaternions $p, q \in \mathbb{H}$ qualssevol, podem tenir $pq \neq qp$. Demanarem a l'usuari els dos quaternions, escriurem pq i qp i direm si coincideixen o no amb una tolerancia donada (10^{-10}). Comproveu també que

   i) $\mathbf{ijk} = -1$.

   ii) $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$.

Definim ara

- el conjugat d'un quaternió $q = q_0 + \mathbf{u} \in \mathbb{H}$ com $q^* := q_0 - \mathbf{u} \in \mathbb{H}$,
- la norma d'un quaternió com $\|p\| = \sqrt{p \cdot p^*}$; si $p = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k} \in \mathbb{H}$ tenim, com a $\mathbb{R}^4$:

$$\|p\| = \sqrt{p \cdot p^*} = \sqrt{p_0^2 + p_1^2 + p_2^2 + p_3^2}\ .$$

Un cop definida la norma, podem definir la divisió de quaternions o l'invers d'un quaternió. L'invers d'un quaternió $p \in \mathbb{H}$ s'escriu p^{-1} i és tal que $p^{-1}p = pp^{-1} = \mathbf{1}$, on $\mathbf{1}$ representa l'element neutre del producte de quaternions, és a dir, $1 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$. D'aquesta manera, multiplicant pel conjugat p^* a les dues bandes obtenim que

$$p^{-1} = \frac{p^*}{\|p\|^2}.$$

c) Escriviu les següents funcions, que reben un quaternió (podeu guardar-les al fitxer **oper-quater.c**):

- `conjQuat` retorna el seu conjugat,
- `normQuat` retorna la seva norma,
- `inversQuat` retorna el seu invers fent servir les funcions anteriors.

Programeu una funció `main` (que guardareu a **mainInvers.c**), que demana un quaternió p, l'escriu, calcula la seva norma i el seu conjugat i els escriu, comprovant que $\|p\| = \sqrt{p \cdot p^*}$. Calcularà la inversa de p, l'escriurà, comprovant que efectivament tenim $p \cdot p^{-1} = 1$.

Per exemple, si $p = 1 + 2\mathbf{i} + 3\mathbf{j} + 4\mathbf{k}$, la sortida del programa serà

```
El quaternio p =
 1.00000000e+00   2.00000000e+00   3.00000000e+00   4.00000000e+00
Amb norma  5.47722558e+00
Te conjugat p* =
 1.00000000e+00  -2.00000000e+00  -3.00000000e+00  -4.00000000e+00
Comprovacio norma: p p* =
 3.00000000e+01   0.00000000e+00   0.00000000e+00   0.00000000e+00
L'arrel de la part real es =  5.47722558e+00
Te inversa p^-1 =
 3.33333333e-02  -6.66666667e-02  -1.00000000e-01  -1.33333333e-01
Comprovacio inversa: p p^-1 =
 1.00000000e+00   0.00000000e+00   0.00000000e+00  -2.77555756e-17
```

Una propietat dels quaternions és que poden representar i operar rotacions en l'espai $\mathbb{R}^3$ mitjançant la conjugació entre quaternions. Definim la conjugació d'un quaternió $q \in \mathbb{H}$ sobre un altre quaternió $p \in \mathbb{H}$ com el quaternió resultant de pqp^*.

Un vector $\mathbf{u} = (u_1, u_2, u_3) \in \mathbb{R}^3$ es pot identificar amb el quaternió $q = 0 + \mathbf{u} = 0 + u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$. Si prenem $\mathbf{d} \in \mathbb{R}^3$ de norma 1 i $0 < \alpha < \pi$, podem definir el quaternió de rotació

$$p = cos(\alpha) + sin(\alpha)\,\mathbf{d}.$$

La conjugació de q sobre el quaternió de rotació p, és a dir, pqp^*, és un altre vector de $\mathbb{R}^3$ (un quaternió amb part real nul·la), que correspon a una rotació de q d'angle 2α amb l'eix de rotació el vector unitari $\mathbf{d}$.

d) Feu una funció `escriuQuat` que, donat un quaternió i un punter a fitxer, escriu en el fitxer la part real, la part imaginària i la norma del quaternió en la mateixa línia.

e) Feu una funció `rotacioQuat` que rep un nombre enter n, el nombre de rotacions a fer, un quaternió inicial (amb part real nul·la), un vector de direcció $\mathbf{d}$, una tolerància `tol` i un punter a fitxer. Donat el quaternió inicial, haurà de fer n rotacions d'angle π/n amb eix de rotació $\mathbf{d}$ com s'ha indicat anteriorment i escriurà al fitxer el quaternió resultant usant la funció `escriuQuat` per cada rotació que es faci. En cada rotació s'ha de comprobar si la part real és zero usant la tolerància indicada. Si no ho és retornarem 0, en cas que tot sigui correcte retornem 1.

f) Feu una funció `main`, que guardareu en el fitxer **mainRotacio.c**, que demana el nom d'un fitxer d'entrada i d'un de sortida. Del fitxer d'entrada anirem llegint, n, el nombre de rotacions que farem, un quaternió q i una direcció $\mathbf{d}$ fins que l'arxiu s'acabi. Per a cada parella $(q, \mathbf{d})$, escriurem al fitxer de sortida les n rotacions que anem fent i deixarem dues línies entre una i l'altre.

   Preneu com exemple el següent arxiu d'entrada

```
100
0 0 1 1
1 0 0
0 1 0 1
0 1 0
0 1 1 0
0 0 1
```

Feu servir el programa **gnuplot** per veure l'evolució de les rotacions. Si el fitxer de sortida es diu `rotacio.out` la instrucció adient seria `splot 'rotacio.out' u 2:3:4 w l`.