

Olimpiada Informàtica Espanyola

Regional Illes Balears



Universitat
de les Illes Balears

31 Gener 2025

Quadern de problemes

Llistat de problemes:

- A. Por el amor de esa mujer
- B. Tití em va preguntar, Amor o Amol?
- C. Joc dels cavalls
- D. Errors en matemàtiques
- E. Les pàgines del diccionari
- F. El laberint
- G. La ruta més eficient
- H. Paraules encadenades
- I. Capturar o no capturar?

Autors dels problemes:

- Ramón Moreno Macarrilla
- Dylan Luigi Canning García
- Denis Molnar Ardelean
- Toni Fuentes Bauzá
- Miquel Oliver Alcayde
- Jose David Pérez Cañellas
- Aina Tur Serrano
- Jaume Cañellas Sanz
- Daniel García Vázquez
- Jose Maria Buades Rubio



A. Por el amor de esa mujer

En David i n'Alex eren dos homes amb un mateix destí, tots dos estimaven la mateixa dona i, encara que fossin amics, no s'anaven a donar per vençuts. Un horabaixa van conèixer uns altres dos homes que també eren amics i estaven enamorats d'una altra mateixa dona. Després d'aquest descobriment, es van plantejar quantes dones tenien més d'un enamorat.

Entrada i sortida

L'entrada consisteix en un primer nombre N que simbolitza el nombre de vegades que els dos amics es van fer aquesta pregunta. A continuació, hi apareixen N nombres X que representen la quantitat d'homes enamorats. La sortida consisteix que, per cada X , ens digui la quantitat de dones estimades, suposant que, com a màxim, només dos homes estaran enamorats de la mateixa dona. Això vol dir que, si el nombre d'homes és senar, l'home restant està enamorat d'una altra dona.

Exemple

Entrada:

```
5
2
4
10
3
5
```

Sortida:

```
1
2
5
2
3
```

Restriccions

La quantitat d'homes mai serà superior a 100.000.

Subtasques

1. (12 punts) Funciona per casos amb 100 o menys homes, sense nombres senars.
2. (28 punts) Funciona per casos amb 100 o menys homes, amb nombres senars.
3. (60 punts) Sense restriccions.



IV Olimpíada Informàtica Illes Balears
OI Illes Balears 2025
Tití em va preguntar, Amor o Amol?



Universitat
de les Illes Balears

Escola
Politécnica
Superior

B. Tití em va preguntar, Amor o Amol?

El cantant Bad Bunny vol treure el seu pròxim àlbum. No obstant això, s'enfronta a un problema peculiar: cada vegada que intenta escriure les seves cançons amb l'ordinador, les tecles *R* i *L* fallen, i intercanvien misteriosament les seves funcions.

Bad Bunny no podia permetre's deixar els seus versos en aquest estat; en cas contrari, els seus fans podrien deixar de ser-li fidel i començarien a seguir altres artistes, fins i tot de gèneres musicals completament diferents. No obstant això, després de llegir els desastrosos versos de les seves cançons detingudament, es va adonar d'una cosa inesperada: l'única paraula amb l'error que mantenia una rima perfecta i aconseguia transmetre amb claredat l'emoció desitjada era *Amor*, que apareixia escrita com a *Amol*. Per això, ha decidit que *Amor* i qualsevol altra paraula que la contengui com ara *enamorar*, *amorf* o *Zamora* han de romandre amb l'error.

A més, a Bad Bunny a vegades li agrada alternar majúscules i minúscules en les lletres de les seves cançons, com en l'exemple següent:

VurR donaL AmoL i fERiCiTat

i el vers corregit seria:

VuLL donaR AmoL i fELiCiTat

Ajuda el cantant Bad Bunny a corregir la lletra de les seves cançons perquè pugui interpretar-les correctament.

Entrada i sortida

L'entrada consisteix en el nombre de casos de prova N , que és la quantitat de frases a corregir. A continuació, cada cas de prova consisteix en una línia de caràcters alfabètics amb espais. Per simplificar, no hi apareixen caràcters amb accents ni caràcters especials.

La sortida consisteix en la frase d'entrada corregida i amb la mateixa longitud de caràcters.

Exemple

Entrada:

```
4
enamolalme
peL a Res envejoses pau i amoL
jo Nomes em DeixO polTal DeR teu somLiuLe
Ra teva LiaRRa es RA miRRoL medecinA
```

Sortida:

```
enamolarme
peR a Les envejoses pau i amoL
jo Nomes em DeixO porTar DeL teu somRiuRe
La teva RiaLLa es LA miLLoR medecinA
```

Restriccions

El nombre màxim de caràcters en una línia és 80.

Subtasques

1. (4 punts) Les frases no contenen errors, per la qual cosa no es fan canvis, tot en minúscules.
2. (13 punts) Totes les L errònies es corregeixen a R , tot en minúscules.
3. (13 punts) Totes les R errònies es corregeixen a L , tot en minúscules.
4. (35 punts) Les frases contenen errors en totes dues lletres (L i R) intercanviades, tot en minúscules, i es corregeixen excepte per a les paraules que contenen *Amor*.
5. (35 punts) Les frases contenen errors en totes dues lletres (L i R) intercanviades. Les paraules estan en una combinació de majúscules i minúscules. El programa ha de corregir els errors i la sortida ha de mantenir el mateix format que l'entrada (espais, majúscules i minúscules).



C. Joc dels cavalls

Els escacs, amb milers d'anys d'antiguitat, són considerats un esport mental per la complexitat a la qual poden arribar les partides: l'estratègia ho és tot per guanyar en aquest joc.

Els escacs es juguen en un tauler de 8×8 caselles numerades d'esquerra a dreta amb les primeres 8 lletres de l'alfabet en majúscula (A-H) i de baix a dalt per números de l'1 al 8.

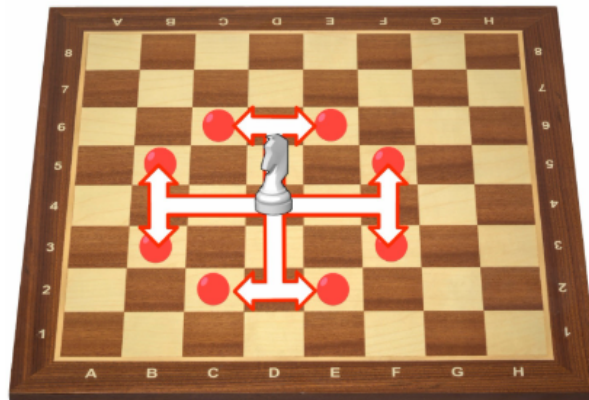
La peça del cavall crida especialment l'atenció per ser l'única que no es desplaça en línia recta, sinó en *L*, així que és fàcil equivocar-se en moure'l. Per aquest motiu, volem fer un sistema que comprovi que els moviments realitzats per un cavall són correctes.

Entrada i sortida

La primera línia conté un nombre N que indica la quantitat de casos de prova.

El cas de prova consisteix en dues línies: la primera conté un número major o igual a 1 que indica quants moviments du a terme el cavall. La segona conté una seqüència de caselles en el format *A1*, separades per espais, que indiquen la seqüència de moviments. La primera casella és on comença el cavall.

La sortida haurà d'indicar la validesa de la seqüència de moviments amb *Valid* o *Invalid*.



Exemple

Entrada:

```
3
1
A1 B3
4
C2 B4 A7 C6 E7
3
F2 G4 E5 F7
```

Sortida:

```
Valid
Invalid
Valid
```

Restriccions

El tauler sempre serà de les dimensions originals 8×8 .

Subtasques

1. (9 punts) El cavall comença a la casella $D4$ i fa un sol moviment.
2. (35 punts) El cavall comença a qualsevol casella i fa un sol moviment
3. (56 punts) El cavall comença a qualsevol casella i pot fer fins a 100 moviments.



D. Errors en matemàtiques

El nostre professor de matemàtiques té una manera particular de corregir els exercicis que han de tenir un resultat numèric. En lloc de puntuar la teva solució, el que fa és anotar quants errors has comès segons la diferència entre els dígitos de la solució correcta i la resposta donada. L'únic que té en compte és que els dígitos de la solució han d'aparèixer a la resposta; per cada xifra que no hi aparegui és un punt de penalització, mentre que per cada xifra que sobri són 5 punts de penalització.

Per exemple, si la solució a un problema és 1224 i la resposta donada és 51205, la penalització és 17: 2 punts perquè hi falten dues xifres (un 2 i un 4) i 15 punts perquè hi sobren tres xifres (dos 5 i un 0).

Ajuda el professor a comptar la penalització de cada resposta.

Entrada i sortida

L'entrada consisteix en el nombre de problemes que ha de corregir el professor.

Cada cas de prova consisteix en quatre línies: la primera línia conté la longitud de la solució; la segona, és la solució del problema de matemàtiques; la tercera línia conté la longitud de la resposta de l'estudiant; i la quarta línia és la resposta de l'estudiant. Ni la solució ni la resposta superen les 100 xifres.

La sortida consisteix en la penalització de l'estudiant.

Exemple

Entrada:

```
3
4
1224
5
51205
4
1234
4
2341
4
1234
1
1
```

Sortida:

```
17
0
3
```

Restriccions

Les solucions i les respostes contindran únicament dígit numèrics.

Subtasques

1. (6 punts) El programa funciona amb entrades de solucions i respostes de longitud 1.
2. (16 punts) El programa funciona amb entrades de solucions i respostes de longitud fins a 2 xifres.
3. (78 punts) El programa funciona amb entrades de solucions i respostes de longitud fins a 100 xifres.



E. Les pàgines del diccionari

N'Abracabassots ha après avui a l'escola a fer servir el diccionari. Li han explicat com estan ordenades les paraules. També li han ensenyat que en un diccionari estan escrites a les cantonades superiors de cada pàgina la primera i la darrera paraula que apareixen en aquella pàgina, per així poder trobar qualsevol paraula molt més ràpidament.

N'Abracabassots està encantat amb aquest descobriment i ha decidit posar en pràctica aquesta tècnica amb la seva amiga Miralunars amb un nou i divertit joc: na Miralunars dirà una paraula a l'atzar i n'Abracabassots ha d'endevinar a quina pàgina del diccionari es troba, consultant només les paraules de la cantonada de cada pàgina.

Entrada i sortida

Aquest és un problema interactiu. Has de refrescar la sortida cada vegada que imprimeixis dades (`cout << endl` o `cout << flush` en C++, `System.out.flush()` en Java, `stdout.flush()` en Python).

L'entrada comença amb una línia amb un nombre enter N que indica el nombre de pàgines que té el diccionari, un nombre enter M amb el nombre màxim de consultes que es poden fer i una cadena de caràcters amb la paraula que ha de trobar n'Abracabassots.

Per fer una consulta sobre la pàgina n del diccionari ($1 \leq n \leq N$), el programa escriurà una línia a la sortida estàndard amb el format `? n` i llegirà una línia de l'entrada estàndard amb dues cadenes de caràcters separades per un espai, que representen la primera i la darrera paraula que apareixen en aquella pàgina, respectivament.

Per donar una resposta amb el nombre de pàgina r on es troba la paraula que cercam, s'escriurà una línia amb el format `! r`. Aquesta interacció no compta com una de les M consultes que es poden fer amb la màquina.

Exemple

Entrada:

```
3 3 diccionari
java python
abella jas
```

Sortida:

```
? 2
? 1
! 1
```

Restriccions

Totes les paraules són en minúscules i tenen, com a molt, 20 caràcters. Només es fa servir l'alfabet anglès, per tant, no hi apareixeran caràcters com la 'ñ', 'ç' o vocals amb accents.

Es garanteix que la paraula a cercar no es troba entre la darrera paraula d'una pàgina i la primera paraula de la pàgina següent, és a dir, sempre es trobarà dins una pàgina del diccionari.

Subtasques

1. (7 punts) $1 \leq N = M \leq 10$.
2. (33 punts) $1 \leq N = M \leq 1000$.
3. (60 punts) $1 \leq N \leq 10^5$; $M = 100$.

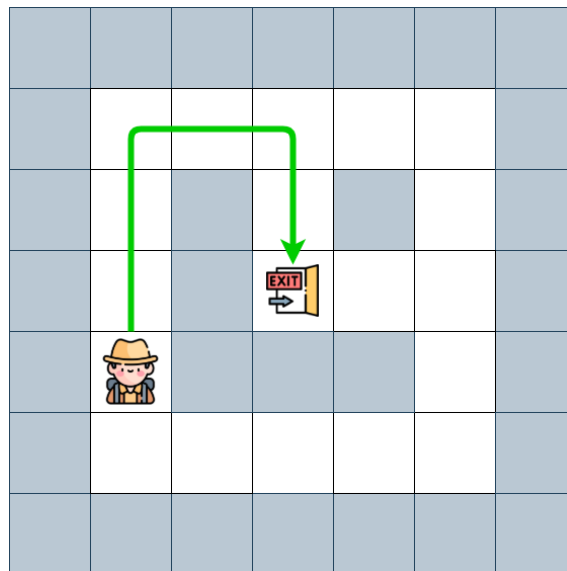


F. El laberint

Segons la mitologia grega, Dèdal va construir un laberint per amagar-hi el minotaure. El nostre benvolgut arqueòleg i explorador Indiana Jones està investigant aquest mite. Per a la seva sorpresa, el mite és real, existeix un laberint.

Afortunadament, el minotaure no és real, però el laberint en si mateix és un parany. Les parets cauen i camins que abans eren transitables ara ja no ho són. El laberint s'està enfonsant!

Ajuda n'Indiana a trobar el camí més curt cap a la sortida del laberint. Si és que realment pot escapar-ne...



Entrada

La primera línia conté un número N que indica la quantitat de casos de prova.

Cada cas de prova té una línia inicial amb dos números H i W , que indiquen l'altura i l'amplada del laberint, respectivament.

A continuació es descriu el laberint en H línies, cadascuna de W caràcters. Els possibles caràcters són:

- X indica que hi ha una paret (no es pot travessar).
- S (start) indica la posició inicial de l'explorador.
- F (finish) la posició de la sortida.
- Un punt (.) indica que és una zona transitable.

Sortida

La solució de cada cas de prova ha d'imprimir-se en una línia diferent, que ha de ser o bé un número que indica la quantitat mínima de passes que ha de fer l'explorador per aconseguir sortir del laberint, o bé el text *IMPOSSIBLE SORTIR* si no es pot trobar cap camí cap a la sortida.

Exemple

Entrada:

```
3
3 5
XXXXX
XS.FX
XXXXX
7 7
XXXXXXX
X.....X
X.X.X.X
X.XF..X
XSXXX.X
X.....X
XXXXXXX
3 7
XXXXXXX
XS.X.FX
XXXXXXX
```

Sortida:

```
2
7
IMPOSSIBLE SORTIR
```

Restriccions

- H i W sempre són major o iguals que 3 i menor o iguals que 100.
- Les vores dels laberints sempre seran X .
- Tots els laberints tenen una única S i F .

Subtasques

1. (11 punts) Els laberints tenen un únic camí possible cap a la sortida, amb H i W menor o iguals que 10.
2. (37 punts) Els laberints poden tenir diversos camins cap a la sortida o no tenir-ne cap, amb H i W menor o iguals que 10.
3. (52 punts) Els laberints no tenen restriccions addicionals.



G. La ruta més eficient

N'Armando Guerra Segura, estudiant d'Enginyeria Informàtica a la UIB, arriba sovint tard a la seva classe d'Anàlisi de Dades a les 8:30 a causa de problemes de trànsit. Per solucionar-ho, ha decidit desenvolupar una aplicació mòbil basada en una API de la DGT de Mallorca, que proporciona informació sobre rutes, velocitat permesa, distància i retard pel trànsit. L'aplicació calcula la millor ruta perquè n'Armando arribi a classe al més a prop possible de les 8:30 i evitar també arribar massa prest per no haver d'esperar innecessàriament.

Entrada i sortida

L'entrada del problema conté una primera línia amb el nombre de casos a analitzar. Després, per a cada cas de prova:

1. Hora de sortida en format `hh:mm` (24 hores), compresa en el rang de 06:00 a 08:20.
2. Nombre de rutes N ($0 \leq N \leq 1,000$).
3. Per a cada ruta, una línia amb els atributs següents separats per espais:
 - *StrCodi*: codi de la carretera (cadena de caràcters).
 - *tempsNormal*: temps normal en minuts per arribar des de casa seva a la Universitat per aquesta carretera (enter).
 - *timeDelay*: retard en minuts a causa del trànsit (enter).

La sortida per a cada cas de prova ha de ser o bé el codi de la carretera de la ruta que permeti n'Armando arribar al més a prop possible de les 8:30, o bé, si cap ruta li permet arribar a temps, el text `NO ARRIBA`.

Exemple

Entrada:

```
2
06:00 4
MA-01 35 45
MA-11 45 75
MA-19 60 85
MA-30 55 93
08:20 2
MA-01 75 3
MA-10 40 1
```

Sortida:

```
MA-30
NO ARRIBA
```

Restriccions

Els atributs compleixen les condicions següents: $StrCodi \in \text{String}$, $tempsNormal, timeDelay \in \text{Natural}$.

Subtasques

1. (10 punts) Pot comprovar 200 carreteres.
2. (15 punts) Pot comprovar 400 carreteres.
3. (75 punts) Pot comprovar 1000 carreteres.



H. Paraules encadenades

En Biel i en Xesc solien jugar al joc de les paraules encadenades sempre que podien. Hi varen jugar tant que es tornaren boníssims jugant i ara les seves partides duren hores i hores, moltes vegades acaben sense que cap dels dos guanyi per culpa d'una interrupció externa. A ningú li agrada un joc en el qual no hi ha guanyadors i, per solucionar aquest problema, han decidit canviar les normes del joc.

Normalment, el joc consisteix en jugadors que, per torns, diuen paraules reals, de manera que només es poden dir paraules que comencin per la lletra amb què acabava la paraula anterior. Per exemple, si el primer jugador escull *pare*, el segon jugador podria escollir *elefant*, però no *animal*. No es poden fer servir paraules que ja s'han emprat i perd el primer jugador que és incapaç de continuar el joc. El primer jugador pot triar la paraula que vulgui en el seu primer torn.

Com en Biel i en Xesc pràcticament s'han après de memòria el diccionari de tant jugar, han decidit limitar les paraules que poden emprar, i utilitzar qualsevol llista de paraules que trobin, sense importar el significat de les paraules. Per exemple, si estan en un restaurant, podrien jugar emprant les paraules de la carta, o si estan en un aeroport, podrien limitar les paraules als noms dels diferents vols.

En aquesta variant del joc, no té avantatge el jugador que conegui més paraules, la té el jugador que esculli paraules de la millor manera possible.

Com hi ha moltes menys paraules, les partides sempre acaben amb un guanyador. En Biel i en Xesc s'han adonat que, si es coneix la llista de paraules, es pot determinar quin jugador guanyarà la partida si tots dos fan els millors moviments possibles. Per exemple, si la llista de paraules és {*aigua*, *aura*}, sempre guanyarà el segon jugador, sense importar les decisions que faci el primer jugador.

En Biel i en Xesc t'han demanat que escriguis un programa que sigui capaç de determinar el guanyador d'una partida, donada la seva llista de paraules. Així es poden riure l'un de l'altre quan qualcú perdi una partida que hauria d'haver guanyat.

Entrada i sortida

L'entrada comença amb un nombre N que representa el nombre de partides que s'han d'analitzar.

A continuació apareixen N casos de prova. Un cas de prova comença per una línia amb un nombre X major que 0, que representa el nombre de paraules de la llista de paraules del cas de prova. A la línia següent hi apareixen les X paraules, separades per espais i escrites en lletres minúscules de l'alfabet anglès (sense accents ni 'ñ' o 'ç').

Per a cada cas de prova s'ha d'imprimir el nom del jugador que pot garantir la seva victòria. Si és el primer jugador, s'ha d'imprimir **Biel**, si és el segon, **Xesc**. Les respostes han d'estar en línies diferents.

Exemple

Entrada:

```
5
2
sebas sopes
1
tomatiga
2
pilota aigua
3
menjar eminem riure
2
pluja emerit
```

Sortida:

```
Xesc
Biel
Biel
Biel
Biel
```

Restriccions

La longitud màxima de les paraules és de 10 caràcters.

Subtasques

1. (1 punt) Funciona per casos amb llistes de paraules d'1 sola paraula ($X = 1$).
2. (6 punts) Funciona per casos amb llistes de paraules de 2 o menys paraules ($X \leq 2$).
3. (17 punts) Funciona per casos amb llistes de paraules de 4 o menys paraules ($X \leq 1$).
4. (76 punts) Funciona per casos amb llistes de paraules de 20 o menys paraules ($X \leq 20$).

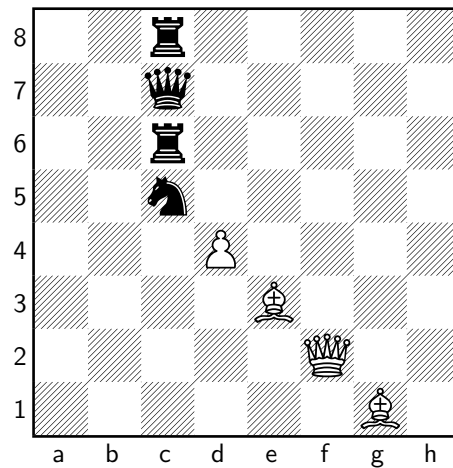


I. Capturar o no capturar?

Els escacs són un esport complex en el qual existeixen molts factors per determinar quin jugador té avantatge en una posició. Un d'ells és l'avantatge material, en el qual s'assigna un determinat nombre de punts a cada tipus de peça. Aleshores el jugador amb una major suma de punts entre totes les seves peces té l'avantatge material.

Molts jugadors aficionats basen els seus càlculs en aquesta puntuació a l'hora de determinar si és rendible capturar una peça i guanyar alguns punts o si, al contrari, la peça pot ser recapturada i acaben perdent punts.

Recentment, la famosa pàgina web d'escacs *chess.com* ha estat treballant en els seus bots d'escacs mensuals que seran publicats en el pròxim mes de febrer. Quan falta un dia del llançament, el bot és capaç de detectar qualsevol peça enemiga amenaçada i determinar les peces atacants i defensores que cobreixen la seva posició. L'únic problema és que el darrer bot encara és incapaç de determinar si capturar la peça pot resultar profitós per guanyar avantatge material quan és el seu torn i està jugant amb blanques o si, al contrari, el pot fer perdre avantatge material.



Com que la resta del teu equip de desenvolupament de la pàgina web *chees.com* està massa ocupat ara mateix jugant a partides ràpides a 5 minuts els uns contra els altres, t'han encomanat a tu acabar el bot per demà.

Entrada

L'entrada comença amb una línia amb el nombre t de casos de prova que vendran a continuació.

La primera línia de cada cas de prova conté tres nombres enters n, m, k ($1 \leq n, m \leq 5 \cdot 10^5$, $1 \leq k \leq 10^9$) que indiquen el valor k d'una peça negra que es troba amenaçada, el nombre n de peces blanques que l'ataquen i el nombre m de peces negres que la defensen.

La segona línia conté n enters $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) amb els valors de les peces blanques que ataquen la peça vulnerable. Les peces actuen una darrere l'altra: la primera peça blanca pot capturar i, si és recapturada per una de negra, la següent blanca pot intervenir-hi, i així successivament.

La tercera línia conté m enters $b_1, b_2, \dots, b_m$ ($1 \leq b_i \leq 10^9$) amb els valors de les peces negres que defensen la peça vulnerable. Les peces negres també actuen una darrere l'altra i segueixen la mateixa estratègia que les blanques: si veuen que recapturant poden acabar perdent material, elles aturaran l'atac.

Com és habitual a les partides d'escacs, les blanques mouen primer i són les primeres que han de determinar si volen capturar o no. En qualsevol moment tant les blanques com les negres poden aturar l'atac i decidir no recapturar per evitar perdre material.

Sortida

Per a cada cas de prova, s'escriurà una línia amb un d'aquests tres missatges:

- dos signes d'exclamació (!!) si capturar la peça és profitós per a les blanques (és a dir, acaben guanyant avantatge material),
- dos signes d'interrogació (??) si capturar causa una pèrdua d'avantatge material per a les blanques,
- si capturar resulta indiferent per a les blanques (no es guanya ni es perd avantatge material), s'escriurà !?.

Exemple

Entrada:

```
3
4 3 3
1 3 9 3
5 9 5
3 4 3
9 3 3
3 5 9 3
2 2 1
1 9
3 5
```

Sortida:

```
!!
??
!?
```

Subtasques

1. (4 punts) $n = m = 1$.
2. (34 punts) $1 \leq n, m \leq 1000$.
3. (62 punts) $1 \leq n, m \leq 5 \cdot 10^5$.