

# Solucions 1r Concurs d'Entrenament OICat 2021

## Olimpíada Informàtica Catalana

Aquí expliquem molt esquemàticament les solucions dels problemes. Al Github hi trobareu els codis. Recordeu que a la [web de l'OICat](#) hi trobareu links a diferents recursos que us poden ser útils, incloent una introducció a Python i a C++, i un tutorial per resoldre problemes gràfics amb Python.

### Problema Q1: Mitjons

Clarament podeu tenir tres mitjons de colors diferents, i pel principi de les caselles, si teniu quatre mitjons, almenys dos seran del mateix color. La solució, per tant, és  $\boxed{4}$ .

### Problema C1: Suma de quadrats

No necessiteu res més que un bucle senzill que vagi sumant els valors de  $k^2$  a la variable on guardeu la resposta.

**Repte:** Podríeu demostrar que  $1^2 + 2^2 + \dots + n^2 = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}$ ? Podeu fer-ho per inducció. (Si algú no sap què és una demostració per inducció, és un bon moment per aprendre-ho.)

### Problema G1: Carta

És un problema fàcil d'implementació. Recordeu, com sempre, de no pintar píxels de més. Si algú encara no sap com implementar un problema gràfic, al web de la OICat hi ha un link a un tutorial sobre el tema.

## Problema Q2: Eleccions

**Autor:** Xavier Povill

Cal escollir un president d'entre les 7 persones amb estudis superiors, i d'entre les 11 persones restants, n'hem d'escollir 2 vocals, cosa que podem fer de  $\binom{11}{2}$  maneres.

Per tant, la solució és  $7 \times \binom{11}{2} = \frac{7 \times 11 \times 10}{2} = \boxed{385}$ .

## Problema C2: Le crguassang

La complicació principal d'aquest problema està en buscar una forma raonable de fer les substitucions. Cal anar en compte també amb algun cas especial, com poden ser les paraules `r`, `ng` o `oi`. El codi C++ que trobareu a Github servirà de millor explicació que el que es pugui posar aquí. A part, com a curiositat també hi veureu una solució en Python d'“una sola línia”, fent ús de les eines de manipulació d'strings que dona el llenguatge.

## Problema G2: Dues diagonals

Itereu per cada fila o per cada columna. Si no sou a la fila o columna central, dibuixeu els dos quadrats corresponents, i si hi sou, dibuixeu un únic quadrat amb el “color mitjana”.

## Problema C3: Barres (3)

Podem veure que el patró per a  $n > 1$  consisteix en dos cops el patró per  $n - 1$ , seguit d'una línia amb  $n$  asteriscs. Això ho podeu aconseguir molt fàcilment amb una funció recursiva.

## Problema C4: Cerca dicotòmica

Si l'interval on hem de buscar és buit, retornem  $-1$ . Altrament, calculem el “punt mig” entre *esq* i *dre*:  $m = \lfloor \frac{esq + dre}{2} \rfloor$ . Si  $v[m] = x$ , el resultat és  $m$ . Si  $v[m] > x$ , busquem recursivament a l'interval  $[esq, m - 1]$ , i si  $v[m] < x$ , busquem recursivament a l'interval  $[m + 1, dre]$ .

**Repte:** Intenteu demostrar que si  $n$  és la longitud inicial de l'interval on es busca, aquesta funció serà cridada com a molt  $\lceil \log_2(n + 1) \rceil$  cops.

## Problema G3: Tauler o esfera?

Una manera de resoldre aquest problema, bàsicament d'implementació, és usar un bucle dins d'un bucle per iterar sobre cada casella, i tenir una funció per decidir, donada una casella  $(x, y)$ , quins dels seus quadradets cal pintar. Per a cada casella, hi pinteu el fons primer, i després els quadradets corresponents. Fixeu-vos que els colors venen donats per la paritat de  $x + y$ , com en un tauler d'escacs.

La complicació més gran del problema consisteix a trobar tots els subcasos per decidir quins quadradets pintar. Fixeu-vos que tots ells depenen de la posició de la casella respecte a la columna i fila centrals. Per a un d'aquests casos, us pot ser útil la noció de **distància Manhattan**.

Com sempre, al Github hi trobareu una implementació.

## Problema C5: Pac-man

Ens guardem dues matrius de booleans, amb les mateixes dimensions del mapa. La primera d'aquestes matrius ens dirà si una casella és visitable, i l'altra, si hi té un bolet. Marquem els obstacles, les caselles amb fantasmes, i les vuit caselles al voltant de caselles amb fantasmes com a no visitables.

Ara, fem un DFS des de la posició inicial del Pac-man, i només deixem que explori les quatre caselles adjacents a la casella actual, sempre i quan siguin visitables. Si trobem una casella amb un bolet, ja podem parar.