Q1.

Para este problema, ajustamos la puntuación del comecocos en función de una serie de criterios, considerando que $c \in C \mid c = (c_x, c_y)$ son las comidas, $s \in S \mid s = (s_x, s_y)$ son los estados del Pacman, $g \in G \mid g = (g_x, g_y)$ son los fantasmas no asustados.

1) Para cada comida, calculamos su distancia con el estado actual del comecocos, de tal forma que $d(s, c) = \min(|s_x - c_x| + |s_y - c_y|)$
2) Para cada fantasma no asustado, calculamos su distancia con el estado actual del Pacman, de tal forma que $d(s, g) = \min(|s_x - g_x| + |s_y - g_y|)$
3) Si $S = \{\emptyset\}$, entonces se ha acabado el algoritmo.
4) Si $G = \{\emptyset\}$, no existirá ningún fantasma asustado, por lo que **no hay penalización**
5) La puntuación de la comida siempre será calculada por su recíproco, es decir, y se penalizará según la posición más cercana del fantasma: $score = (score + (d(s, c) + 1)^{-1}) - [10 * (d(s, g) + 1)^{-1}]$

Q2.

Para este problema, ajustamos la elección de acción del comecocos en función de una estrategia minimax, considerando que $s \in S \mid s = (s_x, s_y)$ son los estados del comecocos, $g \in G \mid g = (g_x, g_y)$ son los fantasmas, y $A = \{a_1, a_2, \dots, a_n \; donde \; n \in \mathbb{N}\}$ son las acciones legales. Pacman, el cual está indicado como $agentIndex = 0$, buscará maximizar la función evaluación, mientras que los fantasmas, indicados como $agentIndex \geq 1$, buscarán minimizar la función. La función se define de forma recursiva como:

$$V(s, i, d) = \begin{cases} f(s), & si\ s\ es\ terminal\ o\ d\ alcanza\ la\ profundidad\ máxima \\ \max\limits_{a \in A} V(succ(s, a), i + 1, d), & si\ i = 0 \\ \min\limits_{a \in A} V(succ(s, a), i + 1, d), & si\ i \geq 1 \end{cases}$$

Donde $succ(s, a)$ es el estado sucesor tras la acción a, y la profundidad d se incrementa únicamente cuando el turno vuelve a Pacman. La acción óptima se elige como:

$a^* = \arg\max_{a \in A} V(succ(s, a)1,0)$.

Q3.

Para este problema, implementamos un agente Alpha-Beta que selecciona la acción óptima del Pacman $s \in S \mid s = (s_x, s_y)$, considerando un conjunto de fantasmas $G = \{g_1, \dots, g_n\}$ y las acciones legales $A = \{a_1, \dots, a_n\}$, donde $n \in \mathbb{N}$. El agente emplea poda α-β para reducir el numero de estados explorados, manteniendo los valores del Minimax.

Para cada estado sucesor $suc(s, a)$, Pacman (recordemos que $agentIndex = 0$) calcula:

$$v = \max_{a \in A} V(succ(s, a), \alpha, \beta, i + 1, d\}$$

Mientras que los fantasmas (Recordemos que $agentIndex \geq 1$) calculan:

$$v = \min_{a \in A} V(succ(s, a), \alpha, \beta, i + 1, d)$$

Con poda: Si $v > \beta$ para Max ∨ $v < \alpha$ para Min, donde se descartarán los sucesores restantes, actualizándose los limites como $\alpha = \max(\alpha, v)$ y $\beta = \min(\beta, v)$, y la acción óptima se escoge obteniendo valores idénticos a Minimax, explorando menos estados.

$$a^* = \arg\max_{a \in A} V(succ(s, a), -\infty, +\infty, 1,0), \qquad \text{[Acción óptima]}$$

Q4

Para este problema, modelamos la toma de decisiones del Pacman mediante el algoritmo **Expectimax**, considerando $s \in S$ los estados del juego, $A = \{a_1, \dots, a_n\}$ las acciones legales, $G = \{g_1, \dots, g_n\}$ los fantasmas.

Pacman ($AgentIndex = 0$) maximiza el valor esperado del estado, definido como:

$$V(s) = \max_{a \in A} V(succ(s, a))$$

Mientras que cada fantasma actúa de manera aleatoria uniforme, calculando la esperanza matemática:

$$V(s) = \sum_{a \in A} \frac{1}{|A|} V(succ(s, a))$$

El algoritmo finaliza cuando el estado es terminal, o se alcanza la profundidad máxima, devolviendo la función de evaluación $f(s)$. La acción más optima se selecciona, considerando el comportamiento probabilístico de cada fantasma como:

$$a^* = \arg\max_{a \in A} V(succ(s, a))$$

Q5

Para este problema, definimos una función de evaluación avanzada que ajusta el estado del comecocos $s \in S \mid s = (s_x, s_y)$ considerando el conjunto de comidas: $F = \{f_1, \dots, f_n | f = (f_x, f_y)\}$ y los fantasmas $G = \{g_1, \dots, g_n\}$, diferenciando entre peligrosos y asustados.

La distancia mínima a la comida se calcula de la siguiente manera:

$d_f = \min_{f \in F}\big(d_{manhattan}(s, f)\big)$, $d(s, f) = |s_x - f_x| + |s_y - f_y|$, con foodParameter $= \frac{1}{d_f + 1}$

Y la distancia a los fantasmas peligrosos como:

$$d_g = \min_{g \in G_{peligrosos}} \big(d_{manhattan}(s, g)\big), d(s, g) = |s_x - g_x| + |s_y - g_y|, dangerGhostParameter = \sum_{g \in G_{peligrosos}} \frac{1}{d_g + 1}$$

Y por último, el parámetro de los fantasmas asustados:

$$scaryGhostParameter = \sum_{g \in G_{asustados}} \frac{1}{d_g + 1}$$

y un ajuste por la comida restante:

$$restFoodParameter = \frac{1}{|F| + 1},$$

Donde finalmente, calcularemos el valor de score, empleando la siguiente ecuación:

$$score = baseScore + \sum_{i=0}^{4} w_i \cdot parametro_i$$

Donde w_i son los pesos que hemos asignado para cada parámetro (dependiendo de la gravedad), y $parametro_i$ es cada parámetro que hemos calculado. Cabe recalcar que el único peso que cambia en función de un estado determinado, es el del fantasma peligroso, si hay fantasmas muy cerca del comecocos, el valor del peso aumenta drásticamente.