

# Hackathon'23

ESI CR - UCLM

6/Noviembre/2023

## El enigma de los piratas (10 puntos)

Tras una larga aventura, un determinado número de aguerridos piratas consiguen llegar a la isla del tesoro. Allí se encuentran enterradas nada más y nada menos que 100 monedas de oro. Parecía que lo difícil ya había pasado, cuando entraron en fuertes discusiones de como repartir el botín. Dado que no se ponían de acuerdo, no les quedó otro remedio que recurrir al código de los piratas. El código de los piratas dice:

Para repartirse un botín, primero el pirata de mayor rango, el capitán, ha de proponer una distribución. Todos los piratas, incluido el que propone, tienen que votar si aceptar o rechazar dicha distribución. Si el plan es aceptado por una mayoría de piratas, ese es el reparto que se hace. Si hay empate, también el reparto es aceptado. Sin embargo, en el caso de que la mayoría lo rechace, el pirata que lo ha propuesto será echado a los tiburones sin piedad alguna. Si eso pasa, el siguiente pirata con mayor rango será entonces el que proponga un nuevo reparto. Este proceso se repetirá hasta que un reparto se acepta o solo quede un pirata (que se llevará todo el botín).

Suponemos que los piratas en la ronda  $n$ , conocen la propuesta que harían sus compañeros en la ronda  $n+1$ . Tu tarea será realizar un programa que sea capaz de tomar como entrada los posibles repartos que cada uno de los cinco piratas haría si les tocará el turno de repartir el dinero y devolverá como salida el reparto final obtenido por cada uno de los piratas, o si estos han sido echados a los tiburones. Se asume que tanto la entrada como la salida están ordenadas de pirata de mayor rango a menor rango.

Un ejemplo de entrada será el siguiente, donde cada línea representa el reparto que sería propuesto por cada pirata si llegara su turno (notar que en cada reparto se asume que el pirata anterior ha sido echado a los tiburones, por lo tanto, se debe repartir entre menos piratas):

```
55 0 45 0 0 0
45 40 2 12 1
40 38 15 7
35 35 20
90 10
100
```

En este ejemplo tenemos 6 piratas. En la primera ronda el primer pirata propone quedarse con 55 monedas, si votara en contra sentenciaría su propia muerte, por lo tanto, su voto será a favor del reparto. El segundo pirata obtiene 0 monedas en este reparto, pero si votara en contra, en el siguiente reparto (que realizaría él mismo) obtendría 50 monedas, por lo tanto, votará en contra. El tercer pirata, obtiene 45 monedas, mientras que en el segundo reparto obtendría solo 40, por lo tanto, este pirata votaría a favor. Respecto los piratas 4, 5 y 6, estos votarían en contra porque la ganancia en el siguiente reparto les sería más favorable. De esta forma, tendríamos 2 votos favorables y 4 en contra, y el pirata de mayor rango sería echado a los tiburones (que representaremos con una X en el reparto).

En la segunda ronda solo quedan 5 piratas. El pirata de mayor rango volvería a votar a favor, ya que si no sería echado a los tiburones. El siguiente pirata votará a favor, ya que en la siguiente ronda su ganancia sería idéntica (40 monedas) y por lo tanto ambas rondas le son favorables. Finalmente, los piratas 3, 4 y 5 votarían en contra ya que en la siguiente ronda ganarían más monedas. De esta forma, el resultado de la votación es 2 votos favorables y 3 en contra, y por lo tanto el pirata es echado a los tiburones.

En la tercera ronda, el primer pirata vuelve a votar a favor por el mismo motivo que los anteriores, el segundo pirata votaría también a favor, mientras que los piratas 3 y 4 votarán en contra. Tenemos 2 votos a favor y 2 votos en contra, por lo tanto, se acepta el reparto.

El **informe generado** sería el siguiente:

X X 40 38 15 7

Como se puede observar, los dos primeros piratas han sido echados a los tiburones, por lo tanto, se representará con una X, mientras que los piratas 3, 4, 5 y 6 obtendrán 40, 38, 15 y 7 monedas respectivamente.

**Se pide:** Implementar el algoritmo capaz de resolver este problema para cualquier fichero de entrada *01.in*, de forma que como resultado se obtenga un *fichero 01.out* con la solución correcta. Ejecuta dicho algoritmo sobre todos los ficheros de entrada proporcionados.