**Второ домашно** – мин-мкс процедура с алфа бета отрязване:

**Дадена е игра за двама играчи**: В началното състояние има N монети, които са подредени последователно в кръг. Монетите с индекс 0 и N-1 се считат за съседни.

Играчите се редуват. На всеки ход, един от тях трябва да вземе 1, 2 или 3 монети, които са съседни (намират се една до друга). Монетите не сменят позициите си по време на играта. Когато една монета е взета от дъската, позицията и остава празна, но продължава да разделя останалите монети.

Печели този играч, който вземе последната монета. Играта не позволява равенство. Играчите се стремят на спечелят в най-малък брой ходове. Ако ще загубят – целта е да е с най-много ходове. Колкото по-бързо спечелите – толкова повече точки ще получите, колкото по-бързо загубите – толкова повече точки ще загубите.

Напишете програма, която препоръчва печеливш ход на играчите, предполагайки, че  и опонентът им ще играе оптимално.

Входът е едно текущо състояние на играта. При стартиране от конзолата се чете N – бройката монети в началото.

Програмата трябва да върне оптималния ход и точките, които играчът ще спечели при оптимална игра и на двамата играчи. Изходъд се състои от 3 числа:

i – позицията, от която играчът ще започне да взема монети.

count – бройката монети, които играчът трябва да вземе на този ход. [1, 2 или 3]

moves – броят точки, които играчът, който е на ход ще спечели. Ако опонентът му ще спечели, стойността на score е отрицателна.

**Примери:**

playCoinGame 3 -> {0, 3, 2} ; Аз вземам всички монети и печеля в един ход.

playCoinGame 4 -> {0, 1, -2} ; Аз вземам първата монета, той взема осаналите.

playCoinGame 5 -> {0, 1, -1} ; Аз вземам първата монета, той взема средните две, аз втората и за него остава последната. Играта завършва в 4 хода и той печели само една точка.

**Упътване за решаване:**

Направете помощна функция, която приема повече аргументи и извиква рекурсивно себе си, за да провери последващите ходове. Ще търсим реализация на пълен min-max (без ограничаване в дълбочина) с алфа-бета отрязване. Примерни аргументи:

n – големината на списъка.

int[] – списък от N елемента с 0-и и 1-ци. 0 означава, че монетата на този индекс вече е била взета от дъската. 1-ца означава, че монетата още е в игра.

moves – бройката ходове, които са направени, за стигане до тази ситуация на играта. Когато оценявате точките на края на играта те могат да бъдат [n – moves] за победа, и съответно (-1)\*[n-moves] зa загуба.

atTurn – кой е на ход – максимизиращ или минимизиращ. Това може да се разбере и от бройката на ходове до сега – четно или нечетно число, но нека не разчитаме на това ☺.

min – достъгнат до сега минимум точки

max – достигнат до сега максимум точки