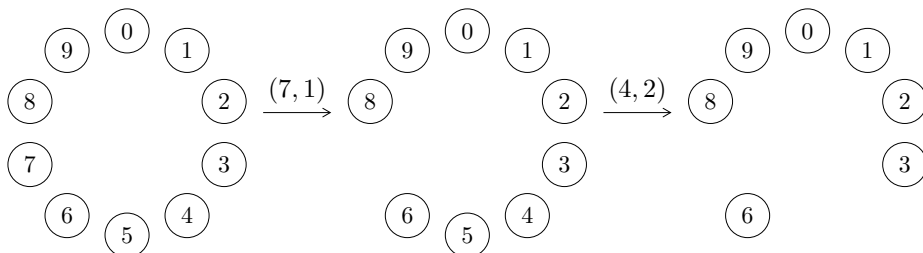


### Taak 1.1 – Stukken (coins) – (100 pts)

We plaatsen  $n$  geldstukken in een cirkel op de tafel en nummeren hen van 0 tot  $n - 1$ , in wijzerzin. Vervolgens spelen we een spel met twee, elk om de beurt (speler 1 begint). Elke beurt neemt de speler die aan de beurt is één of twee geldstukken weg van de tafel. Als de speler kiest er twee te nemen, moeten ze aan elkaar grenzen. De winnaar van het spel is degene die het laatste stuk wegneemt.

Hier is een voorbeeld van het begin van een spel. Het eerste getal komt overeen met de index van het eerst genomen stuk (in wijzerzin). Het tweede getal geeft het aantal genomen stukken aan.



#### Taak

Voor deze taak moet je de strategie voor speler 2 implementeren zodat die telkens kan winnen.

#### Limieten en beperkingen

De grootte  $n$  ligt tussen 10 en 100 (inclusief), bovendien:

Subtaak A (25pts)	Het aantal stukken is even, speler 1 speelt willekeurig
Subtaak B (25pts)	Het aantal stukken is even, speler 1 speelt op een hoger niveau
Subtaak C (25pts)	Het aantal stukken is even, speler 1 speelt optimaal
Subtaak D (25pts)	Het aantal stukken is oneven, speler 1 speelt optimaal

De score wordt als volgt berekend: er zijn 25 verschillende spelletjes in elke subtaak. Je krijgt 1 punt per spel dat door speler 2 gewonnen wordt.

#### Te implementeren functie

`int play(int i, int last_coin, int last_nb, int size):` Deze functie neemt als argumenten:

**i** De nummer van de beurt, je krijgt dus 2, 4, 6, ...

**last\_coin** De index van de stukken die net door speler 1 genomen zijn. Als hij twee stukken genomen heeft, is het de index van het eerste stuk, wijzerzin.

**last\_nb** Het aantal stukken dat speler 1 net genomen heeft: 1 of 2.

**size** Het aantal stukken aan het begin van het spel.

Deze functie met de zet die speler 2 zal spelen teruggeven in de vorm van een integer, gebruik de functie `int playValue(int coins_index, int coins_nb)` om die aan te maken. Deze neemt als argumenten:

**coins\_index** De index van de te nemen stukken. Als je 2 stukken kiest, is het de index van het eerste stuk, wijzerzin.

**coins\_nb** Het aantal stukken dat je wegneemt: 1 of 2.

## Voorbeeld

We beschouwen het onderstaande voorbeeld.

Hier volgt een mogelijk spel:

Je ontvangt	Je geeft terug	Stuk(ken) speler 1	Stuk(ken) speler 2
<i>play</i> (2, 7, 1, 10)	<i>playValue</i> (4, 2)	7	4, 5
<i>play</i> (4, 2, 2, 10)	<i>playValue</i> (8, 2)	2, 3	8, 9
<i>play</i> (6, 0, 2, 10)	<i>playValue</i> (6, 1)	0, 1	6

In dit geval wint speler 2 omdat hij het laatste stuk heeft weggenomen.

## Praktische opmerkingen

Deze taak is interactief. Je zal dus niet de automatische compilatie en test van gedit kunnen gebruiken voor deze taak. Volg de onderstaande stappen om je programma manueel te compileren en te testen. Vraag hulp aan een toezichter indien nodig.

- Open een terminal (de applicatie LXTerminal)
- Om de lijst met bestanden in een folder weer te geven, gebruik het commando `ls`
- Om van folder te veranderen, naar het skelet van deze taak, gebruik `cd folder_naam`

Voor degenen die C++ gebruiken, zijn de commando's de volgende:

- Compilatie: `g++ -std=c++11 -Wall grader.cpp coins.cpp`
- Uitvoering: `./a.out < input.txt`

Voor wie Java gebruikt, zijn de commando's als volgt:

- Compilatie: `javac grader.java`
- Uitvoering: `java grader < input.txt`

Het resultaat van jouw programma zal afgeprint worden in de console: "Player lost" als je (speler 2) het spelletje verloren hebt en "Player won" als je gewonnen hebt. Als je het bestand `input.txt` wil aanpassen, moet je weten dat dat het aantal stukken, een *seed* voor de willekeurigheid van speler 1 (verander die om een ander spel van speler 1 te krijgen) en niveau van speler 1 bevat (maar enkel het willekeurig spel is gegeven in de `grader` die jij krijgt om de optimale strategie niet prijs te geven).

## Opmerkingen

- Het is verboden een functie op te roepen om willekeurige getallen te genereren om deze taak op te lossen.
- De oplossing is vrij kort. Als je een uitgebreide spelstrategie aan het schrijven bent, ben je waarschijnlijk op het verkeerde pad.
- Je moet enkel het bestand `coins.cpp` of `coins.java` indienen. Je moet de functie `play` implementeren, zoals hierboven beschreven.
- Als je na het indienen "Resultaat incorrect" krijgt, wil dat typisch zeggen dat je verloren hebt.

- Als je na het indienen krijgt dat de statuscode verschillend is van nul, wil dat typisch zeggen dat je een actie hebt uitgevoerd die niet volgens de regels van het spel is.
- Je mag er uiteraard van uit gaan dat speler 1 altijd volgens de regels zal spelen.
- Print niets uit naar de console in het programma dat je indient.