

Objetivos

- Aplicación de los conocimientos adquiridos durante el curso
- Integración del alumno a un grupo de trabajo, en el entendimiento que ***el trabajo en grupo propicia el análisis, la discusión y la aplicación de lo aprendido***

Características del trabajo obligatorio

- Los grupos deben tener **3 integrantes**. En caso de que la totalidad de alumnos no sea múltiplo de 3, se admitirá excepcionalmente algún grupo de 2 integrantes.
- La fecha de entrega será el día **Jueves 03/08/2017 hasta las 23:00 hs.**
- El trabajo obligatorio vale 100 puntos y representa el 25% de la calificación final del curso. Se debe obtener un mínimo de 60 puntos para aspirar a la aprobación del curso. El puntaje obtenido en el obligatorio es **grupal**.
- Se debe enviar un archivo comprimido (.zip) conteniendo los archivos fuente de la entrega a la dirección de correo electrónico indicada por el docente. Indicar en el cuerpo del mail los nombres y cédulas de los integrantes del grupo.
- El trabajo consiste en la implementación de la solución al problema que se plantea a continuación. Todos los integrantes del grupo deben participar en las discusiones y en la resolución de los diferentes algoritmos propuestos.

Planteo del problema

Se trata de un juego de apuestas en el cual van a participar **M** jugadores. Al comenzar el juego, se van a registrar los nombres de los jugadores y a cada uno de ellos se le van a entregar **K** monedas iniciales para apostar, las cuales utilizarán para realizar apuestas durante las (como mucho) **N** rondas que se realizarán durante el juego. En cada ronda, cada jugador apostará la cantidad de monedas que desee (de entre aquellas monedas que tiene disponibles) a un número que elegirá entre 0 y **C** - 1. Tras las apuestas de la ronda, se sorteará uno de esos números al azar. Aquellos jugadores que hayan apostado al número ganador obtendrán una ganancia neta igual al doble de la cantidad de monedas que apostaron. Por ejemplo, si un jugador apostó 3 monedas al número ganador, recibirá 6 monedas en total (recuperará las 3 monedas apostadas y recibirá 3 adicionales). Los jugadores que no hayan apostado al número ganador, perderán las monedas apostadas en la ronda. Culminada la ronda, se mostrará en pantalla la cantidad de monedas que tiene cada jugador tras culminar la ronda. El juego finalizará en el momento en que alguno de los jugadores se quede sin monedas o cuando hayan transcurrido las **N** rondas (lo que pase primero). Finalizado el juego, se mostrará por pantalla el nombre del ganador junto con su cantidad final de monedas. El ganador del juego será el que tenga la mayor cantidad de monedas al momento de finalizar. En caso de haber empate, se declarará ganador a cualquiera de ellos. Se pide hacer un programa en lenguaje C que declare los valores para **M**, **K**, **N** y **C** como constantes simbólicas y que realice las siguientes acciones:

1. **Registro de jugadores:** Se ingresarán por teclado los nombres de los jugadores, los cuales se almacenarán en un arreglo de **M** posiciones, una para cada nombre. El contenido de cada celda será un string conteniendo el nombre del jugador. Puede ocurrir que haya más de un jugador con el mismo nombre, por lo que se permitirá el ingreso de nombres repetidos. Por ejemplo, a continuación se muestra un posible arreglo para **M** = 3.

0	1	2
"Benito"	"Ursula"	"Petrona"

Se llevará la cuenta de la cantidad de monedas que tiene cada jugador en todo momento del juego por lo que, además del arreglo de nombres, se tendrá un segundo arreglo (también de **M** celdas) conteniendo la cantidad de monedas que tiene cada jugador. Este arreglo se cargará con el valor **K** en todas sus celdas al inicio del programa.

2. **Realización de rondas durante el juego:** Una vez registrados los **M** nombres de jugadores y cargadas las **K** monedas iniciales de cada uno, se procederá a la realización de las rondas. Se jugarán, como mucho, **N** rondas, excepto que algún jugador se quede sin monedas antes, en cuyo caso el juego terminará en ese momento. En cada ronda, se ingresará por teclado el número apostado por cada jugador y la cantidad de monedas apostadas. El programa deberá controlar que el número apostado esté efectivamente entre 0 y **C** - 1 y que la cantidad de monedas no supere el total de monedas que tiene el jugador, para lo cual solicitará que se reingresen cualquiera de los dos valores (tantas veces como sea necesario) hasta garantizar que fueron ingresados correctamente. Esta información se almacenará en una matriz de apuestas, de tamaño **M** x **C**. Por ejemplo, para **M** = 3 y **C** = 5, se muestra a continuación la matriz de apuestas al momento de realizar las apuestas de la primera ronda:

	0	1	2	3	4
0	0	2	0	0	0
1	0	0	0	4	0
2	0	3	0	0	0

El primer jugador (Benito) apostó 2 monedas al número 1. El segundo jugador (Ursula) apostó 4 monedas al número 3. El tercer jugador (Petrona) apostó 3 monedas al número 1. Cada jugador puede apostar solamente un número en cada ronda, cuya celda correspondiente en la matriz de apuestas se cargará con la cantidad de monedas apostadas, mientras que el resto de las celdas de la fila de dicho jugador se cargarán siempre con ceros. No se pide mostrar la matriz de apuestas por pantalla, alcanza solamente con utilizarla en memoria para registrar las apuestas. Tomando **K** = 10 como cantidad inicial de monedas, se muestra a continuación el estado del arreglo de monedas tras realizadas las apuestas de la primera ronda. Benito quedó con 8 monedas, Ursula quedó con 6 monedas y Petrona quedó con 7 monedas.

0	1	2
8	6	7

Acto seguido, se procede al sorteo del número ganador de la ronda, el cual será elegido al azar entre 0 y **C** - 1 (se debe investigar cómo generar un número al azar en el lenguaje C) y mostrado por pantalla. Si alguno de los jugadores quedó sin monedas al terminar la ronda, el juego terminará en ese momento. Como ejemplo, vamos a suponer que el número sorteado es el 3. Esto significa que Ursula gana 8 monedas (recupera las 4 apostadas y gana otras 4 adicionales), mientras que Benito y Petrona pierden las monedas apostadas. Se muestra a continuación cómo queda el arreglo de monedas, cuyo contenido será desplegado por pantalla tras culminada la ronda:

0	1	2
8	14	7

3. **Finalización del juego:** Culminadas las (como mucho) **N** rondas, se declarará ganador al jugador que haya quedado con la mayor cantidad de monedas tras culminadas las rondas. En caso de que haya más de uno, se declarará ganador a cualquiera de ellos. Como resultado, se mostrará por pantalla el nombre del ganador junto con su cantidad final de monedas y el programa finalizará su ejecución.