

## Trabajo 1

## Programación 2

Docente: Ing. Elina Avila Ordóñez

### Indicaciones:

- Este trabajo corresponde a una actividad evaluada sobre 12 puntos del Aprovechamiento 1
- Se realiza en los grupos establecidos, pero la nota es individual
- Deberá subirse al e-virtual hasta el **8 de noviembre hasta las 08:00** (Subir Trabajo 1). Entregas tardías se penalizarán con 1 punto hasta los 30 minutos, con 7 puntos hasta los 180 minutos y con 12 puntos para más de 180 minutos. Se subirá como archivos comprimidos. Y OBLIGATORIAMENTE se debe colocar como nombre de este archivo: GRUPOn. Donde n es el número de su grupo. Si su trabajo **no tiene** esta nomenclatura será penalizado con 1 punto.
- Solo 1 integrante del grupo sube el trabajo. Si hay 2 o más entregas de un mismo grupo se califica la primera.
- El trabajo debe desarrollarse en **C no en C++** y con el IDE DEV C++. Si no se cumple este requisito se penalizará el trabajo con 1 punto.
- Lea detenidamente el enunciado y desarrolle lo solicitado.
- Recuerde que Usted ha aprendido todos los conceptos necesarios para desarrollar este trabajo. ¡Éxitos!

### ENUNCIADO:

#### Juego de Pares



El famoso juego de pares es un divertimento muy sencillo de plantear y ejecutar. Se basa en una matriz que deberá tener un número de celdas pares. Cada celda contiene un valor que deberá obligatoriamente aparecer 2 veces en la matriz. Luego, un participante selecciona dos celdas y esperará hallar un par. Cada vez que atine a un par obtendrá 1 punto, caso contrario no acumulará puntaje. El número de intentos será limitado de la siguiente forma: el número de celdas de la matriz dividido para dos más el 30%. Los participantes podrían alcanzar el puntaje máximo en un juego que es igual al número de

celdas de la matriz dividido para dos. Los puntajes obtenidos se almacenarán en un ranking de forma que se pueda presentar los participantes de mejor a peor calificados.

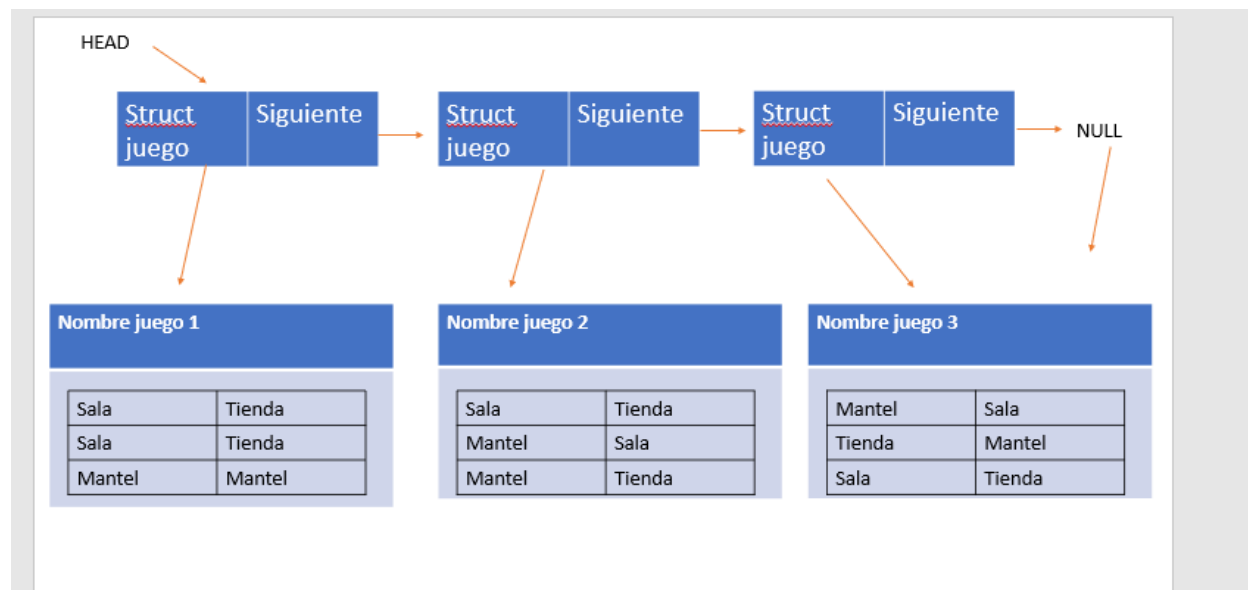
### Detalles del software

El Software deberá contar con un menú como el siguiente:

1. Crear juego
2. Crear participante
3. Jugar
4. Mostrar ranking
5. Datos para probar
6. Salir

#### *Opción 1: Crear juego*

El software debe permitir crear juegos, es decir, podremos tener muchas matrices con distintas palabras ubicadas con una distribución diferente para jugar. Las matrices guardarán palabras en sus celdas, por lo que los participantes deberán hacer pares de palabras. Todas las matrices del juego tendrán igual dimensión. Se deberá contar con la lista ligada JUEGOS que tendrá nodos que cómo dato tienen un puntero a una estructura tipo juego. Un juego tendrá el nombre del juego, por ejemplo: Juego 1 y la matriz con los datos dentro. El software permitirá el ingreso de juegos sin límites, pues la lista JUEGOS es dinámica. Antes de iniciar, se deberá preguntar al usuario la dimensión que tendrán las matrices. En la siguiente figura se proveen detalles sobre esta estructura:



La lista Juegos (en la parte superior de la figura) estará en el HEAP, y los juegos (parte inferior de la figura) en el STACK. Note que la matriz del juego es una matriz puramente estática. El software deberá validar que los juegos sean válidos.

### *Opción 2: Crear participante*

Para jugar el juego, el participante debe estar creado en el sistema. Para ello el software solicitará su nickname. No pueden existir participantes con el mismo nickname. Un participante además tendrá su puntaje, que al crearlo será 0. Los participantes se guardarán en un vector estático llamado Participantes que residirá en el Stack.

### *Opción 3: Jugar*

Una vez creados juegos y participantes, se podrá jugar. El software pedirá el nickname del participante e iniciará el juego para lo que seleccionará aleatoriamente uno de los juegos previamente creados. Los juegos no deberían repetirse en una partida. Luego, cada participante empezará sus intentos y el software contabilizará puntos y finalizará la partida cuando los intentos se agoten. El resultado se almacenará en el campo puntaje correspondiente al participante en el vector Participantes.

Para simplificar el problema, cuando se ingresa a la opción jugar cada participante deberá participar una vez.

### *Opción 4: Mostrar Ranking*

El software deberá presentar el top de participantes, para ello creará un arreglo estático de igual longitud que el número de participantes. Este arreglo llamado Ranking es un arreglo de punteros a algo de tipo participante. El software deberá procesar los puntajes para llenar el arreglo Ranking de tal manera que la posición 0 del arreglo apunte al participante con mejor puntaje del vector Participantes, la posición 1 al participante con segundo mejor puntaje y así sucesivamente.

Luego recorriendo el vector Ranking, se presentará en la pantalla de mejor a peor calificado los nicknames de los participantes y sus puntajes. Luego de mostrar el ranking los resultados deberán eliminarse y los puntajes de los participantes deberán encerse.

### *Opción 5: Datos para probar*

Para facilitar las pruebas de este software se ha incluido esta opción. Usted deberá crear una función que llene (con hard code) todos los datos requeridos para que el software puede permitir el inicio de una partida.

### *Opción 6: Salir*

Todo lo solicitado al HEAP queda liberado y el programa finaliza.

## **CONSIDERACIONES GENERALES:**

Como todo buen software, este deberá evitar el ingreso de datos erróneos, deberá proveer un menú de opciones de manera que el usuario pueda seleccionar que sección del enunciado ejecutar y deberá mantener informado al usuario de las acciones que se llevan a cabo. Analice cuando una opción del menú puede o no ejecutarse. Por ejemplo, no se podría jugar si no existieran participantes creados.

Al usar punteros los programadores debemos asegurarnos de liberar el HEAP.

El trabajo se realizará en grupos para fortalecer la capacidad de trabajar en equipo de cada uno de ustedes. Forme parte activa del grupo aportando y apoyando a la solución. Intente aplicar buenas prácticas de programación siempre y aplique técnicas como el pair programming.

Para muchos programar es una actividad que resulta mejor si se escucha música el mismo tiempo. Aquí algunas sugerencias para acompañar su proceso:

- Un poco de rock en inglés: <https://www.youtube.com/watch?v=EGa5h0OStUY>
- Para los que gustan de la música tranquila, el polaco Chopin nos ha dejado melodías para piano fantásticas: <https://www.youtube.com/watch?v=-gDinVAmtA0>
- Para cuando vaya quedando poco tiempo, y se necesitamos activar nuestro cerebro: <https://www.youtube.com/watch?v=4PUHBL1vMNY&t=5833s>
- Para los que les gustan los playlists hechas a medida: <https://www.youtube.com/watch?v=WWUoq4RtHV0&t=4076s>

#### EVALUACION:

Este trabajo se evalúa sobre 12 puntos en función de los siguientes criterios:

- Opción 1 del menú: 2.5 puntos (completamente correcta)
- Opción 2 del menú: 2 puntos (completamente correcta)
- Opción 3 del menú: 2.5 puntos (completamente correcta)
- Opción 4 del menú: 2 puntos (completamente correcta)
- Opciones 5 y 6 del menú: 1 puntos (completamente correctas)
- Buen software, buena solución, calidad, completitud y correctitud: 2 puntos

#### DUDAS SOBRE LAS ESTRUCTURAS:

El día martes 31 de octubre se revisarán las estructuras generadas por los grupos. Cada grupo tendrá 5 minutos para mostrar cómo están creando las estructuras solicitadas y podrán recibir algo de feedback. Solo 1 representante del grupo deberá asistir según el horario que se pondrá a disposición oportunamente. Este trabajo se hará en la oficina del grupo MAS del edificio de administración. Favor traer su computador y estar puntuales. Las clases se reemplazan por esta actividad