

Práctica obligatoria final

En esta práctica vamos a implementar un juego de **bingo** un tanto peculiar. Las reglas del juego serán las siguientes:

1. El jugador tendrá un único cartón de 5 x 5 celdas conteniendo números enteros, comprendidos entre el 1 y el 75, que se rellenarán aleatoriamente respetando una serie de restricciones:
 - a) Cada columna solo podrá contener números comprendidos entre los subrangos que se detallan a continuación:
 - B = 1..15
 - I = 16..30
 - N = 31..45
 - G = 46..60
 - O = 61..75
 - b) Los números del cartón no se ordenarán
 - c) La posición central del cartón se considerará acertada en todos los cartones generados

De este modo, el resultado final podría ser similar al que se muestra en esta imagen:

B	I	N	G	O
7	29	43	49	73
2	16	39	53	66
8	19	★	48	74
11	28	33	51	64
9	24	44	59	70

La captura facilitada está tomada de la versión para Android, gratuita en *Play Store*, del juego denominado *Bingo heaven*, aunque los requisitos para su implementación se han modificado en este enunciado que será, en todo caso, **la única especificación que se seguirá para su realización y entrega**.

El sorteo y las apuestas se gestionarán del siguiente modo:

2. El sorteo de números aleatorios se desarrollará de manera que todos los números que se generan en el sorteo aparecerán marcados en una tabla de 5 x 15

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75

El tipo de datos óptimo y recomendado para esta tabla es el siguiente:

```
TCelda = RECORD
    contenido: integer;
    acertado: boolean;
END;
```

3. Cada vez que se genera un número en el sorteo, se tacha en la tabla completa, se muestra por pantalla la tabla actualizada y se le ofrece al jugador la posibilidad de introducir las coordenadas de su cartón en las que aparece dicho número o bien de pasar al siguiente número sorteado
4. Esta operación se repite hasta que produce un bingo, que se entiende como la sucesión de 5 (siendo esta cifra dependiente de la dimensión del tablero) números acertados consecutivamente en sentido vertical, horizontal o en cualquiera de las dos diagonales del tablero.
5. Para que el juego se pueda desarrollar simularemos más jugadores haciendo que el sistema genere un número aleatorio de jugadores (de 1 a 3) y de cartones apostados por cada uno de ellos (de 1 a 4). Estos cartones no se mostrarán por pantalla pero el sistema verificará automáticamente sus aciertos de modo que cualquiera de ellos pueda vencer al jugador.
6. Las apuestas se realizarán del siguiente modo:
 - a) El programa establecerá el mínimo de créditos que el jugador debe apostar
 - b) El sistema, al generar los cartones del resto de jugadores, asignará una cantidad de créditos por cada uno de ellos (también aleatoria) que, junto con la cantidad de créditos aportada por el jugador, formarán el bote que va a ganar el jugador que consiga el primer bingo
7. El programa almacenará la siguiente información del jugador: Nombre completo, créditos, lista de mejores resultados, fecha del último acceso al juego y, opcionalmente, si es la primera vez que entra a jugar.

Se entiende que los mejores resultados son las cinco mejores apuestas del jugador. De los resultados, que solo se guardarán si el jugador ha obtenido una victoria, debemos conocer la cantidad de números acertados en la partida ganada y la cantidad de créditos obtenidos al finalizar dicha partida.

El tipo de datos óptimo y recomendado para estas apuestas es el siguiente:

```
TRecords = RECORD
    numeros_acertados: integer;
    credits_obtenidos: integer;
END;
```

8. La información de los jugadores se almacenará en un fichero de tipo `TJugador`.

El tipo de datos óptimo y recomendado para los jugadores es el siguiente:

```
TCadena = STRING [40];
TRecords = RECORD
    numeros_acertados: integer;
    creditos_obtenidos: integer;
END;
TLista = ARRAY [INI..FIN] OF TRecords;
TFecha = RECORD
    anno: word;
    mes: word;
    dia: word;
    dia_sem: word;
END;
TJugador = RECORD
    nombre: TCadena;
    creditos: integer;
    lista: TLista;
    fecha: TFecha;
    {nuevo: boolean;}
END;
```

9. Respecto a la fecha, será necesario tener en cuenta las siguientes consideraciones:

- Cuando el jugador accede por primera vez al juego o cuando la diferencia entre la fecha almacenada y la actual es superior a un mes, el valor del campo créditos se pone a 25.
- Cada vez que el jugador accede un nuevo día al juego el sistema le bonificará con 5 créditos más.
- Para obtener la fecha del sistema será preciso emplear la unidad `DOS`. En esta unidad está implementado un procedimiento cuya cabecera es la siguiente:

```
procedure GetDate(var year, month, mday, wday: word);
```

10. Para finalizar, en memoria principal, podrá habilitarse un `array` como estructura auxiliar que va a permitir gestionar los rankings de los jugadores de forma que puedan mostrarse por pantalla por orden alfabético o por puntuación, siendo mejor la puntuación en la que se combinen la menor cantidad de números acertados con la mayor cantidad de créditos obtenidos. Será el usuario del programa quien elija el modo de ordenación, con lo que tendréis que implementar ambos.

La práctica solicitada consiste en realizar de forma **grupal** el juego del **bingo** en Pascal utilizando **exclusivamente los tipos de datos que se han explicado en clase y están contenidos en la guía docente de la asignatura** y teniendo en cuenta las siguientes observaciones:

- Para crear el fichero de jugadores o sobrescribirlo cuando ya haya sido creado, será necesario más de un subprograma.
- No es necesario emplear las utilidades `TextColor` y `Delay` de la unidad `CRT`, pero no está prohibido.

CRITERIOS DE EVALUACIÓN

Hacer el juego completo tendrá un valor de 10 puntos.

Será motivo de **suspenso directo** el uso de variables globales sin haberlas pasado como parámetro de los subprogramas en los que se empleen y el uso de tipos de datos no explicados en clase e incluidos en el temario del curso.

Será motivo de penalización con **-1 punto** el uso de más parámetros de los necesarios, es decir, el uso de parámetros que, en realidad deberían ser variables locales al subprograma.

Será motivo de penalización con **-2 puntos** el uso de variables cuando sea más eficiente emplear constantes.

Será motivo de penalización con **-5 puntos** el uso de las instrucciones `exit`, `halt`, `break`, o `continue`.

NORMAS DE ENTREGA DE LA PRÁCTICA

La práctica debe hacerse **grupalmente** y se entregará en un fichero con el nombre **POF_2018_2019_GXX**, representando las X el grupo elegido. El fichero a entregar será enviado solo por el portavoz del grupo, estará comprimido (.ZIP, .RAR o .7Z) y contendrá:

1. `POFbingo.pas`, el fichero con el código fuente del programa
2. `jugadores.bin`, el fichero con la información almacenada de las partidas jugadas
3. `memoria.pdf`, que contendrá todo el código de la práctica con los comentarios pertinentes explicando las decisiones tomadas en la programación del juego y el **nombre completo y número de expediente de los alumnos**

La práctica se podrá entregar hasta las 23:00 del día 11 de enero de 2019