2do. Trabajo Práctico

a) El trabajo deberá realizarse en forma grupal.
b) Cada grupo solo debe desarrollar 3 ejercicios, 1 de cada tema (a elección)
c) Presentar solo código Pascal bajo el nombre TP2-apellido-apellido-apellido.PAS, en aula virtual. donde figure como un comentario en línea de código, al comienzo del programa:

Comisión de cursado - Apellido y nombre de cada integrante

El objetivo de este trabajo práctico es aplicar todos los conceptos dados en la cátedra hasta el momento, pudiendo utilizar todas o algunas de las **cinco estructuras de control**, combinadas con **datos simples y estructurados array**, además de aplicar los conocimientos referentes a: **manejo de pantalla**, **procedimientos y funciones** con sus respectivos **parámetros**.

Para ello el programa presentará un menú de opciones con el siguiente diseño:

1) Juegos
2) Cálculos
3) Código de Barras /QR
4) Fin

1) Cada grupo elegirá el desarrollo de un juego: Ruleta electrónica ó Ahorcado ó TaTeTi ó Batalla Naval.
2) Desarrollar 1 opción entre: Montos expresados en números convertirlos en texto ó Protectores de pantallas ó Calendario Gregoriano.
3) Elegir y desarrollar Código de Barra ó Código QR.

OPCION 1

Ruleta electrónica

Se solicita un programa en Pascal que simule el juego de la ruleta.
Cada partida consta de varias jugadas en las cuales los jugadores van realizando sus apuestas y sumando o restando puntos según ganen o pierdan las apuestas.
El ganador del juego es el que reúna el mayor número de puntos posibles.
Una partida termina con una condición de fin elegida, o cuando los jugadores participantes se queden sin saldo.
Al comenzar la partida, se debe indicar cuantos jugadores habrá (máximo 5).
Cada jugador tendrá 100 puntos para apostar.
La ruleta tiene 36 números (del 1 al 36) dispuestos en 12 filas y 3 columnas (ver figura):

	1	2	3
1	**1**	**2**	**3**
2	**4**	**5**	**6**
3	**7**	**8**	**9**
4	**10**	**11**	**12**
5	**13**	**14**	**15**
6	**16**	**17**	**18**
7	**19**	**20**	**21**
8	**22**	**23**	**24**
9	**25**	**26**	**27**
10	**28**	**29**	**30**
11	**31**	**32**	**33**
12	**34**	**35**	**36**

En cada jugada de la partida, cada jugador puede realizar sólo un tipo de apuesta:
a) **Fila**: apuesta un valor del 1 al 12.
b) **Columna**: apuesta un valor del 1 al 3.

c) **Par-Impar**: apuesta 'P' o 'I'.
d) **Pleno** (un sólo número): apuesta un nro. del 1 al 36.

Validar el ingreso en todos los casos.
Una vez elegido el tipo de apuesta y su valor, se debe ingresar la cantidad de puntos a apostar (siempre que el jugador tenga saldo suficiente para ello). Validarlo. Repetir esto para cada jugador.
En cada jugada, después de que los jugadores han realizado las apuestas, se gira la ruleta y sale un número entre el 0 y el 36. Con el número obtenido se comprueban dichas apuestas. Si el jugador pierde, se le decrementa el número de puntos apostados del saldo, y si gana, se le aumenta según corresponda por el tipo de apuesta. Si el número que sale es 0, la banca gana (todos los jugadores pierden).
En el caso que el jugador gane, se multiplica la cantidad apostada por un coeficiente según el tipo de apuesta:

a) **Fila**: multiplica el importe apostado por 12.
b) **Columna**: multiplica el importe apostado por 3.
c) **Par-Impar**: multiplica el importe apostado por 2.
d) **Pleno** (a un sólo número): multiplica el importe apostado por 36.

Este resultado se incrementa al saldo de puntos del jugador correspondiente.
Después de que cada jugador haya hecho su apuesta y se haya girado la ruleta, se indicarán los jugadores que ganaron (si los hubo) y se mostrará un resumen de sus puntos actualizado.

Nota: hacer girar la ruleta significa generar un nro. aleatorio en cada jugada, dicho nro. aleatorio se informará por pantalla. Para hacerlo utilizar la función *random(x)* de Pascal.
Random genera un nro. aleatorio entero desde 0 en adelante, y x es el tope o nro. máximo a generar.
Antes de random, utilizar además la función *randomize*; que permite generar una nueva secuencia de números aleatorios cada vez que se corre el programa.

TaTeTi

juegan dos jugadores ó un jugador y la PC, sobre un tablero de 9 posiciones y gana el primero que coloca sus fichas en línea.

	X	
		0

Un jugador juega con las X y el otro jugador (PC) con los 0. Se van colocando las fichas de a un jugador por vez y el primero que logra colocar sus fichas en línea (horizontal, vertical o diagonal) gana.
Si se completa todo el tablero y ninguno de los jugadores pudo ganar, hay empate.
Preparar la pantalla con:

- titulo del juego en letras grandes en color
- Nombre jugador 1:… juega con fichas: X
- Nombre jugador 2:… juega con fichas: O
- Y la cuadrícula de juego para iniciar el juego

Finalizado el juego exhibir si hubo algún ganador mostrando el nombre sino mostrar "empate".
Observación: Mostrar la información solicitada con formatos de pantalla a elección (colores, sonido, recuadros, etc.). Debe solicitar el ingreso de información de manera clara. La interacción con el programa debe ser lo más amigable posible (mensajes, limpieza de pantalla entre los ingresos de datos, etc.) Se debe poder jugar todas las veces que se desee.

Ahorcado

La palabra que se debe adivinar no tendrá más de 10 caracteres.
Al ingresar la palabra automáticamente se debe ir borrando (tipo clave) de forma tal que el jugador no pueda leer lo que se escribe en pantalla.
Ejemplo: al escribir la palabra HOLA, debería ir apareciendo de la siguiente forma (todo en una misma fila de la pantalla) ****

```
H
*O
**L
***A
****
```

Al jugar: Por cada carácter que se introduzca que no pertenezca a la palabra, se visualizará en pantalla uno de los siguientes caracteres ‘ O I / \ ’ hasta que se forme la figura del ahorcado, momento en que se habrá perdido el juego. Los caracteres que se adivinen aparecerán en pantalla en la posición que ocupan en la palabra. Trabaja con arreglos uni y bidimensionales. Figura del ahorcado

```
 o
/I\
 I
/ \
```

Batalla Naval

Juegan dos jugadores y gana el primero que hunde todos los barcos del contrario. Este programa debe considerar que el primer jugador ingresa las posiciones de sus barcos sin que el otro lo vea y luego el segundo jugador intenta hundir los barcos del primer jugador.
Cuando el segundo jugador da las coordenadas de sus intentos tiene 3 posibles respuestas del programa: hundido, averiado o agua.
Se debe contar la cantidad de intentos hasta hundir todos los barcos. Se puede considerar que gana el jugador que en una serie determinada (arreglada por ambos jugadores) hunde al contrario con la menor cantidad de intentos. O proponer otra.
Considerar al tablero de 10 filas por 10 columnas, y la cantidad de barcos son:

- 1 de 4 casilleros
- 2 de 3 casilleros
- 3 de 2 casilleros
- 4 de 1 casilleros

							X	X	
	X	X	X	X					
							X		X
	X								X
						X			X
				X					
				X				X	
								X	
X	X	X				X			

OPCION 2

Conversor de número a texto

Ingresando cualquier valor numérico entre 0 y 9999, en una variable NUM: real; exhibir su valor equivalente en palabras, para ser impreso en pantalla.
Ej: ingresando como parámetro el número 1247, exhibir: ‘mil doscientos cuarenta y siete’.
Si se ingresa el número 983,25, exhibir ‘novecientos ochenta y tres con 25/100’.

El NUM debe ser descompuesto en unidades, decenas, centenas y unidad de mil. Los decimales se mantienen numéricos.
Se debe trabajar con arreglos predefinidos, de forma tal que el dígito obtenido pueda ser utilizado como posición relativa del arreglo.
Ej: el 4, debería permitir que decenas[4] exhiba ‘cuarenta’.

Deberán desarrollar un procedimiento/función llamado **PESOS (NUM: REAL),** con un parámetro de entrada.
Observación: Pueden utilizar string para el mensaje. Recordar que para unir string se utiliza el signo +, ejemplo:

```
Pal = 'mil';
Pal = Pal + ' doscientos';
```

Protector de Pantalla

Se pretende que en la pantalla, una vez que el programa comience a ejecutarse, sobre un fondo oscuro se "iluminen" puntos en colores. Se puede trabajar con un arreglo uni ó bidimensional, de forma tal que el programa rescate las coordenadas (fila y columna) y pueda colocar en la pantalla el punto a iluminar.
El proceso debe ser iterativo hasta que el usuario presiona una tecla cualquiera.
Pueden eligir el motivo del protector (usando la tabla de código ASCII).

Calendario Gregoriano

Averiguar cual es el día de la semana en que caerá una fecha cualquiera.
Ejemplo: **27 de mayo de 2014 fue martes**.
Ingresando una fecha en formato día, mes y año, se deberá establecer por cálculos que día de la semana sucedió dicha fecha.
Trabajar con arreglos para días de la semana, donde la posición 1 será lunes.
Trabajar con arreglos para los meses. Tener en cuenta años bisiestos por la modificación del mes de febrero.
Se puede partir de la fecha actual y retroceder a la fecha solicitada.

OPCION 3

Código de barras

(representación gráfica) a partir de un código numérico, este se forma con los valores correspondientes a:

- código de la empresa: número de 3 cifras (representado por 3 dígitos)
- importe: número entero mayor o igual a cero y menor a 10000 (representado por 4 dígitos)
- dígito verificador: se genera a partir de los 7 dígitos correspondientes a los datos anteriores calculando el resto de dividir por 10 el valor absoluto de la diferencia entre la sumatoria de los dígitos que ocupan una posición par y la sumatoria de los que ocupan una posición impar.

Ejemplo:

código de empresa: 312
importe : 870
} *1º parte del código = 3120870*

Determinación del dígito verificador:
S posiciones pares = 1 + 0 + 7 = 8
S posiciones impares = 3 + 2 + 8 + 0 = 13
Dig.verificador = resto de (abs(8-13)/10) = 0
} *Código numérico: 31208700*

Cada dígito del código numérico así formado se convierte a **notación binaria de 4 bits** de forma que cada bit 1 se representa con una barra vertical y cada bit 0 con un espacio en blanco, quedando así formado el código de barra para ser enviado a la impresora.
Para el ejemplo el código de barra corresponderá a la cadena: 0011000100100000100001110000000
y será: 

Dado que no disponemos de impresoras, generar una representación simbólica para pantalla, ejemplo: los espacios en blancos se pueden representar con un * y las barras verticales negras con una I.
Se ingresará por teclado el código de la empresa que es único, y luego ingresar de forma iterativa los importes y mostrar el código de barras, proponiendo un fin de datos acorde.

Código QR

Representación gráfica de 2 dimensiones, donde cada número entre 0 y 9 será representado **en forma binario con 8 bits** (0 – 1).

Del código QR se trabajará con la versión 1, de 21 x 21 bloques (ó codewords), el cual les da la posibilidad de completar 26 codewords (todos los que sobren podrán darle otro uso de codificación para su representación gráfica visual).
Los 3 ángulos de referencia (dibujo comprendido entre la letra c – son espacios de posicionamiento de la cámara), pueden completarlos con un símbolo a modo de representación gráfica.

1) Elegir forma gráfica de representar los 0 y 1 en pantalla
2) Solo deben representar 4 dígitos (ingresados por pantalla) en las posiciones D1, D2, D3, D4.
3) Visualizar planilla de conversión de caracteres imprimibles ASCII (con 8 bits): http://es.wikipedia.org/wiki/ASCII

Ejemplo legajo 1234:

1	2	3	4
0011 0001	0011 0010	0011 0011	0011 0100
D1	D2	D3	D4

1234 5678
(posiciones)

D1 (orden de bits el 1 es el mas significativo)

2	1	0	0
4	3	1	1
6	5	0	0
8	7	1	0