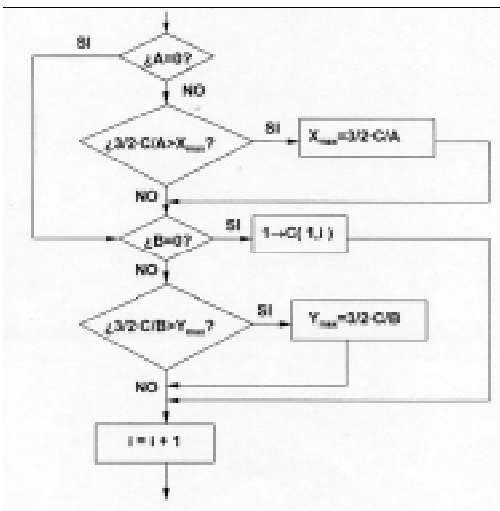


# Introducción a la PROGRAMACIÓN TI 83



T<sup>3</sup> España

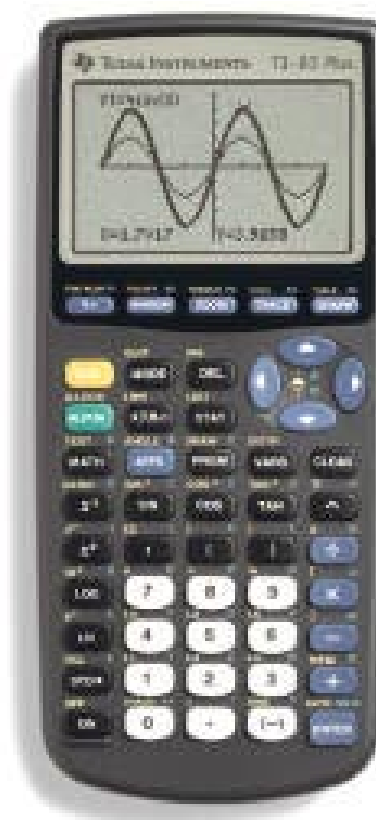


T<sup>3</sup> EUROPE

Tomás Queralt  
Julio Rodrigo  
Salvador Caballero  
Floreál Gracia  
Fernando Juan  
Alfred Molla  
Onofre Monzó  
José Antonio Mora  
Pascual Pérez

## ÍNDICE

1.- Introducción .....	Pág. 3
2.- Simulaciones .....	Pág. 4
3.- Programación lineal .....	Pág. 7
4.- Contraste de hipótesis .....	Pág.15
5.- Bibliografía .....	Pág.20



T3 EUROPE es una marca registrada de Texas Instrumens



## 1.- INTRODUCCIÓN

La programación en las calculadoras gráficas resulta sencilla para la mayoría de los estudiantes de matemáticas por su facilidad. Las ventajas sobre los ordenadores son varias: su simple “sistema operativo”, con el que llegan a familiarizarse, no necesita para ello un almacén de memoria por separado, y su tamaño, que hace que su transporte sea fácil.

Un programa es simplemente un conjunto de instrucciones que están guardadas de forma permanente, con formato muy similar (si no idéntico) a las que puedes escribir directamente. Las instrucciones se guardan en líneas separadas mediante el símbolo de los dos puntos “:”, que indica el principio de una instrucción. A cada programa nos referiremos por un nombre, que será el que lo identifique.

Las ventajas de la programación pueden usarse de tres maneras distintas:

1.- El programa está almacenado en la memoria y los alumnos intentan averiguar lo que el programa hace sin ejecutarlo. (Esto permitiría a los estudiantes explorar situaciones del tipo “entradas a” y “salidas de” en máquinas con funciones simples).

Por ejemplo: Describir qué hace el siguiente programa.

```
ClrList L1  
10→dim(L1)  
For(I,1,60,1)  
randInt(1,10)→Z  
L1(Z)+1→L1(Z)  
End
```

2.- Los estudiantes tienen un programa dado al cual entran por ellos mismos, lo hacen funcionar, lo modifican para hacer otras cosas (por ejemplo, generando secuencias, usando fórmulas simples, observando la forma de gráficas de funciones simples, usando ordenadores para generar figuras bidimensionales, usando  $m$  y  $c$  en  $y = mx + c$ , usando una calculadora para investigar si una secuencia dada por iteración es convergente o divergente, etc.)



3.- Los estudiantes escriben ellos mismos sus propios programas. Como los estudiantes llegan a familiarizarse con la calculadora gráfica, pueden estar motivados para usar las facilidades de la programación cuando sea apropiado.

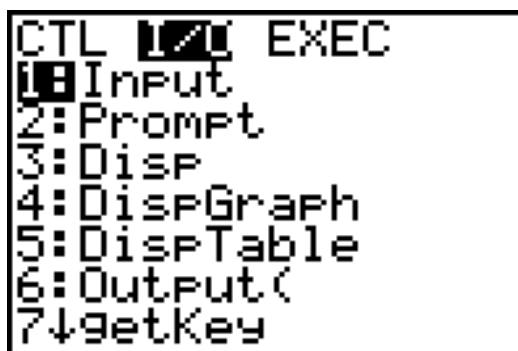
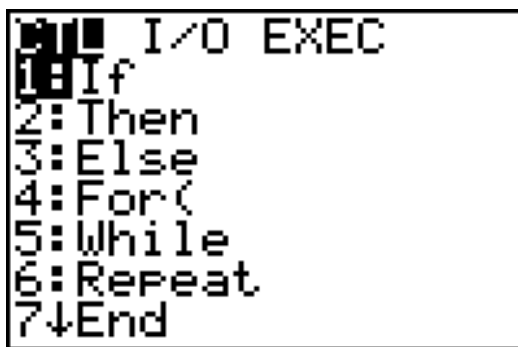
En cualquier caso la corrección de errores en la calculadora es fácil. Las teclas de las filas te permiten moverte a través del programa al lugar donde está el error. Después puedes hacer:

- sobreescribir la instrucción correcta;
- borrarla usando la tecla **DEL**;
- insertar caracteres usando la tecla **2nd [INS]**, y se insertarán antes de la posición del cursor.

## 2.- SIMULACIONES

La utilización de los programas en la calculadora gráfica resulta altamente interesante a la hora de realizar simulaciones, como en el caso de los experimentos aleatorios. Para comenzar vamos a simular el lanzamiento de una moneda, y vamos a representar en un diagrama de barras las frecuencias que se obtengan.

En primer lugar accedemos al menú de programación con la tecla **PRGM**, y yendo a **NEW** crearemos el nuevo programa dándole un nombre tras pulsar **ENTER**. Usando el teclado alfabético, lo llamaremos **MONEDA** (no más de ocho caracteres comenzando siempre por una letra). Para introducir las instrucciones del programa, iremos siempre al menú **PRGM**, y elegiremos aquel submenú que nos interese: **CTL** o instrucciones de control, **I/O** o instrucciones de entrada/salida, y **EXEC** para ejecutar un programa como una subrutina dentro del programa que estamos diseñando.



La primera instrucción será **ClrHome**, como primera medida para limpiar la pantalla cuando se empieza a ejecutar. Cada vez que pulsemos **ENTER** editamos el registro o línea de instrucción y pasamos a una nueva, la cual queda señalada con la aparición de los dos puntos : . Para simular el lanzamiento de una moneda, podemos asignar el valor 0 al resultado cara y un 1 al resultado cruz. Vamos al menú **MATH** **▾** submenú **PRB** y elegimos la instrucción **5** **randInt(** que genera números aleatorios enteros. Su sintaxis es *mínimo, máximo [ , número de simulaciones ]*. Podemos hacer 100 simulaciones, y guardar el resultado en una lista. Por último, pediremos que muestre gráficamente los resultados.

```
PROGRAM: MONEDA
:ClrHome
:randInt(0,1,100
)→L1
:DispGraph
```

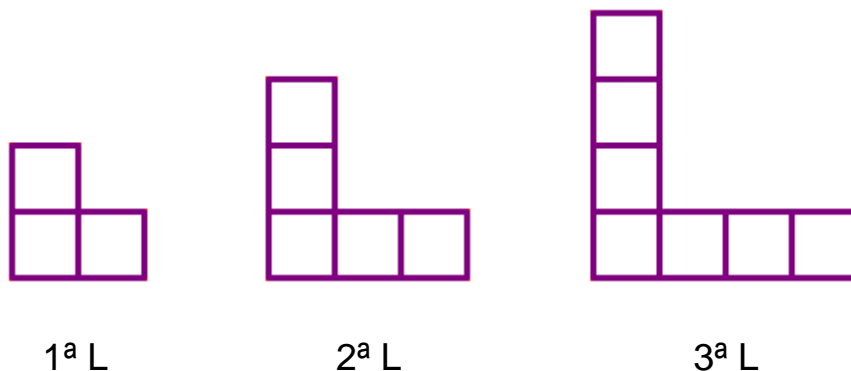
```
WINDOW
Xmin=-1
Xmax=3
Xscl=1
Ymin=0
Ymax=60
Yscl=10
Xres=1
```

```
Plot1 Plot2 Plot3
Off Off
Type: ▽ ▽ ▽
Xlist:L1
Freq:1
```

Solamente faltará ajustar los márgenes de la ventana al tamaño de las barras a dibujar, e indicarle a la máquina que queremos dibujar en un diagrama de rectángulos las frecuencias obtenidas. Ahora ya podemos ejecutar el programa.

**Actividad 2.1.-** Simular el lanzamiento de un dado y representar gráficamente el resultado en un diagrama de barras.

**Actividad 2.2.-**



Predice el número de cuadraditos necesarios para construir la 4ª L, la 5ª L, ... la 20ª L, la 50ª L.

¿Podrías generar esta secuencia usando la calculadora gráfica?



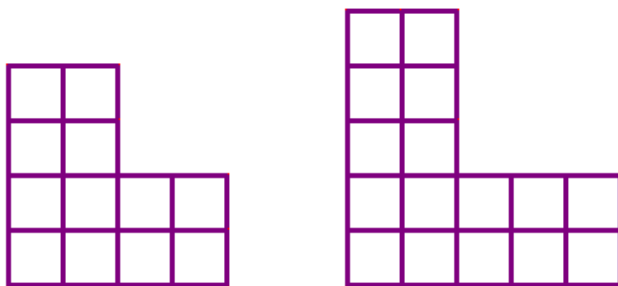
Vamos a crear un programa que me genere dicha secuencia. Con la tecla de **PRGM** iremos al menú correspondiente y le llamamos ELES. Tras poner la instrucción de limpieza de pantalla, inicializamos un contador a cero, el cual irá aumentando de uno en uno para proporcionarnos los valores de la sucesión. Con la instrucción de control **Lbl 1** señalamos el principio de un bucle. Con la instrucción de control **IS>(variable, valor)** vamos a ir aumentando 1 a la variable hasta llegar al valor que nos interese, que será 20 en este caso. La instrucción permite realizar los mandatos que siguen hasta que la variable llega a ser mayor que el valor determinado en la instrucción, caso en que el siguiente mandato se salta. Este mandato sera calcular  $2X+1$ , que es la expresión que me proporciona el número de cuadraditos que tiene la L de la etapa X. Después diremos que nos muestre dicho valor, con la instrucción **DISP ANS**, y posteriormente una pausa para observar dicho resultado. Tras ella, el control debe ser mandado al principio del bucle, con la instrucción **GOTO 1**. El mandato **END** cierra el programa.

```
PROGRAM: ELES
:ClrHome
:0→X
```

```
PROGRAM: ELES
:ClrHome
:0→X
:Lbl 1
:IS>(X, 20)
:2X+1
:Disp Ans
:Pause
```

```
PROGRAM: ELES
:Lbl 1
:IS>(X, 20)
:2X+1
:Disp Ans
:Pause
:Goto 1
:End
```

**Actividad 2.3.-** Supongamos que la forma de las L tienen dos filas de cuadraditos:



1ª L

2ª L

Genera la secuencia de números con la calculadora gráfica para letras con forma de L.

Genera secuencias de números con la calculadora gráfica para otras formas de letras con cuadraditos, como E, F, H, T.



### 3.- PROGRAMACIÓN LINEAL

Hemos visto que elaborar un pequeño programa para la calculadora gráfica es relativamente sencillo, aún cuando no se tiene mucha práctica en la programación con lenguajes de alto nivel. No obstante, antes de empezar a introducir instrucciones, es conveniente hacerse un pequeño esquema de lo que se quiere obtener con su ejecución. En ocasiones, y especialmente si el programa es largo, es recomendable elaborar un pequeño diagrama de flujo, para asegurarse de que se ejecutan aquellas acciones que se pretende, y que no se deja ninguna opción fuera del control.

Como ejemplificación de un programa con un poco más de complejidad, vamos a elaborar un programa que nos permita representar la región factible de un problema de programación lineal. Hay que tener en cuenta lo siguiente:

- Nos vamos a restringir a problemas de programación lineal bidimensional, y por tanto, trabajaremos con dos variables: X, Y.
- Las restricciones se representarán como funciones de una variable respecto de la otra, por lo que se realizará a través del editor de funciones. De esa manera, cada restricción se podrá representar por uno de los dos semiplanos que determina la recta de los puntos que la verifican con igualdad:  $AX+BY \leq C$ . Si  $A=0$ , me aparecerá una recta horizontal, mientras que si  $B=0$  no tengo función, y la restricción es una recta vertical.
- Los coeficientes de la función objetivo y del sistema de desigualdades o restricciones se almacenarán en matrices.

Empezaremos dándole un nombre: **PLGEN**. Las primeras instrucciones serán de control de la pantalla. Así, deberemos desactivar todas las posibles funciones que tengamos en el editor de funciones, para no interferir con nuestra representación. Para ello, pulsaremos **[VARS]**, y en el submenú **Y-VARS** pulsaremos el **[4]**. Con la instrucción **FnOff** desactivamos todas las funciones.



```

VARS Y-VARS
1: Function...
2: Parametric...
3: Polar...
4: On/Off...
  
```



Ahora debemos desactivar también las posibles gráficas estadísticas. Para ello iremos a [2nd] [STAT PLOT] donde **PlotsOff** realiza ésta acción.

```

2:Plots TYPE MARK
1:Plot1(
2:Plot2(
3:Plot3(
4:PlotsOff
5:PlotsOn
    
```

A continuación, activaremos la visualización de los ejes. Para ello, habrá que visualizar la pantalla WINDOW FORMAT: pulsaremos [2nd] [FORMAT] y bajaremos hasta **AxesOn**, pulsando después [ENTER].

```

PROGRAM:PLGEN
:FnOff
:PlotsOff
:AxesOn
: .1→Xmax:.1→Ymax
:ClrHome
:Input "NUMERO R
ESTR.:",N
    
```

Por último, asignaremos unos valores iniciales a los máximos de ventana: pulsando [VARS] iremos al subdirectorio **Window...** También limpiaremos la pantalla con **ClrHome** que se encuentra en el submenú **I/O** del menú de programación.

```

PROGRAM:PLGEN
:PlotsOff
:AxesOn
: .1→Xmax:.1→Ymax
:ClrHome
:Input "NUMERO R
ESTR.:",N
:ClrHome
    
```

A continuación empezamos a solicitar información del problema a resolver. Empezamos por el número de restricciones, el cual nos determinará la dimensión de la matriz de los coeficientes. Así, pulsaremos:

```
:Input "NUMERO RESTR..:", N
```

La instrucción **Input** permite mostrar un mensaje en la pantalla, que lo escribiremos entre comillas, y después de una coma escribiremos la variable donde queremos que se guarde esa información. El texto que aparecerá en la pantalla debes escribirlo desde el teclado alfabético, por lo que hay que pulsar para cada letra la tecla [ALPHA]. Sin embargo, si el texto es largo, se puede fijar el teclado alfabético con [2nd] [A-LOCK]. Para desactivarlo bastará con volver a pulsar [ALPHA].





Tras esta información, vamos a indicarle al usuario que introduzca las restricciones, y la forma que tienen estas. Empezamos limpiando la pantalla con **ClrHome**, y después con la instrucción **Disp** mostraremos los distintos mensajes.

```
PROGRAM:PLGEN
:Disp "INTRODUCE
"
:Disp "RESTRICCI
ONES:"
:Disp "A,B,C PAR
A"
:Disp "AX+BY≤C"
```

Con la instrucción **Output**( podemos colocar un mensaje en la fila y columna de la pantalla que nos interese. Así, con `:Output(6,4,"PULSA ENTER")` nos aparecerá ese mensaje, que seguido de la instrucción **Pause** permitirá detener momentáneamente la ejecución del programa hasta que pulsemos de nuevo la tecla **ENTER**.

Ahora vamos a ir pidiendo los datos, que guardaremos en una matriz. Hará falta previamente dimensionar dicha matriz. Inicializamos un contador:  $1 \rightarrow I$ , y la dimensionaremos teniendo en cuenta que pondremos en cada columna los coeficientes de la restricción correspondiente: entre llaves ponemos el nº de filas (3) y de columnas (N). La instrucción **dim**( aparece en el submenú **MATH** del menú **MATRIX**. Con la instrucción **Fill**( (la cual aparece en el mismo submenú) rellenaremos de ceros la matriz A. También dimensionaremos una matriz vector C, la cual utilizaremos para señalar aquellas restricciones en las que B (coeficiente de la variable Y en las restricciones) valga cero.

```
PROGRAM:PLGEN
:Disp "AX+BY≤C"
:Output(6,4,"PUL
SA ENTER")
:Pause
:1→I
:(3,N)→dim([A])
:Fill(0,[A])
```

```
PROGRAM:PLGEN
:Fill(0,[A])
:(1,N)→dim([C])
:Fill(0,[C])
:Lbl 0
:ClrHome
:Disp "RESTRICCI
ON"
```

Ahora vamos a iniciar el bucle que me permita llenar la matriz de coeficientes. Para volver cada vez a la primera instrucción del bucle, marcamos con una etiqueta el principio: **Lbl 0**. A partir de aquí hacemos las siguientes acciones: limpiamos la pantalla con **ClrHome**, mostramos el número de restricción que vamos a introdu-

```
PROGRAM:PLGEN
:Disp "NUMERO"
:Output(2,8,I)
:Input "A=",A:A→
[A](1,I):Input "
B=",B:B→[A](2,I)
:Input "C=",C:C→
[A](3,I)
```



cir, para que el usuario no se equivoque, e introducimos los coeficientes A, B y C en la columna correspondiente de la matriz A. Podemos unir varias instrucciones sin más que separarlas por los dos puntos ":". Hacer notar que nos referimos al elemento (i,j) de la matriz A poniéndolo entre paréntesis después del nombre de la matriz.

```
PROGRAM:PLGEN
B=",B:B→[A](2,I)
:Input "C=",C:C→
[A](3,I)
:If I=1
:"([A](3,1)-[A](
1,1)X)/[A](2,1)"
→Y1
```

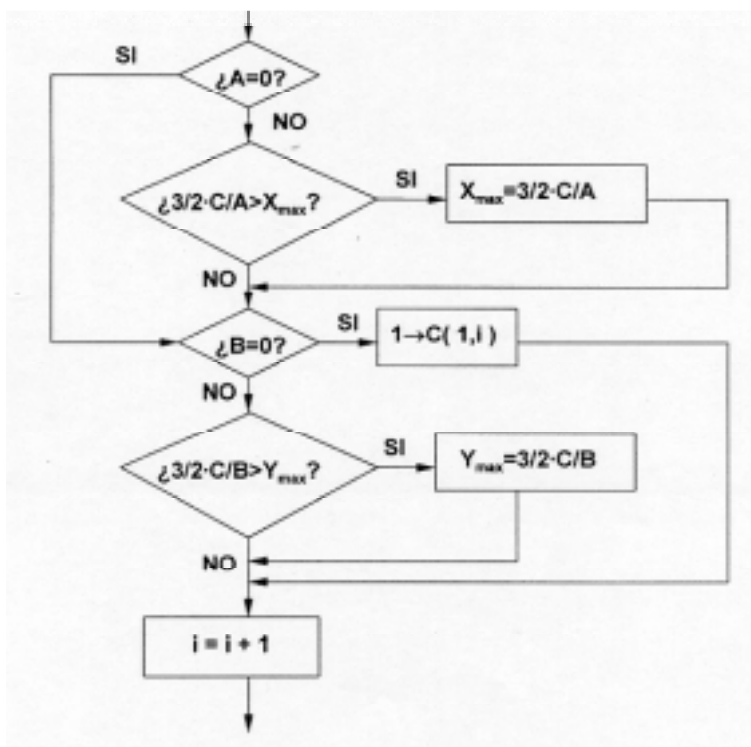
A continuación guardaremos la restricción I en forma de función para poderla representar más tarde. Para ello, habrá que guardar en Y<sub>1</sub> la recta correspondiente a la primera restricción, en Y<sub>2</sub> la correspondiente a la segunda, y así sucesivamente. Así, preguntaremos por el valor del índice I: si I vale 1,

```
PROGRAM:PLGEN
:"([A](3,1)-[A](
1,1)X)/[A](2,1)"
→Y1
:If I=2
:"([A](3,2)-[A](
1,2)X)/[A](2,2)"
→Y2
```

$$Y_1 = \frac{C - AX}{B} = \frac{(A(3,1) - A(1,1)X)}{A(2,1)}$$

y esto hasta el valor 8. Usamos la instrucción **If**, que evalúa la expresión que le sigue, y si es cierta ejecuta la siguiente instrucción, y si no, la salta y pasa a la siguiente. Notar que ponemos entre comillas la expresión que asignamos a cada función con la intención de que se asigne la expresión sin evaluar.

Ahora vamos a ver qué hacemos con los casos especiales, es decir, aquellos en los que algún coeficiente de la restricción es cero.



En primer lugar, ajustaremos el marco de la ventana para que aseguremos de que se dibujan los cortes con los ejes. Eso ocurre en los puntos (0, C/B) y (C/A,0).

Empezaremos preguntando si A vale cero, en cuyo caso trasladaremos el control a una instrucción posterior (donde preguntaremos por el valor de B), y en caso contrario preguntamos si un poco más de C/A supera al  $X_{max}$ . La sintaxis de la instrucción es: **If condición Then instrucción1 Else instrucción2 End**. Si la condición es cierta, se ejecuta la instrucción1 (que puede ser un grupo de instrucciones), y si es falsa se ejecuta la instrucción2. End indica el final del bucle.

Ahora nos preguntamos por el valor de B: si es cero, entonces la restricción se corresponde con una recta vertical, la  $X=C/A$ . En una matriz que llamamos C indicaremos aquella restricción que cumpla esta característica, y mandaremos el control a una instrucción posterior. Si B no es cero, nos aseguraremos que el punto de corte de la recta con el eje Y cae dentro de la ventana. Una vez acabado este bucle pasamos a la siguiente restricción aumentando una unidad el contador.

Una vez han pasado todas las restricciones, sólo falta ajustar los mínimos de la ventana, así como las marcas sobre los ejes.

Después de limpiar la ventana, empezaremos a pedir la función objetivo: pedimos A y B y guardamos en la función  $Y_0$  el caso en que pasa por el origen de coordenadas.

```
PROGRAM:PLGEN
:If A=0:Then
:Goto R
:Else
:If (3/2)(C/A)>X
max
:(3/2)(C/A)→Xmax
:End
```

```
PROGRAM:PLGEN
:Lbl R
:If B=0:Then:1→I
C1(1,I):Goto S
:Else:If (3/2)(C
/B)>Ymax
:(3/2)(C/B)→Ymax
:End
```

```
PROGRAM:PLGEN
:End
:Lbl S
:(I+1)→I
:If I≤N
:Goto 0
:-(Xmax/3)→Xmin
:-(Ymax/10)→Ymin
```

```
PROGRAM:PLGEN
:-(Xmax/3)→Xmin
:-(Ymax/10)→Ymin
:int((Xmax-Xmin)
/20)→Xscl
:int((Ymax-Ymin)
/20)→Yscl
:ClrHome
```



A partir de ahora sólo nos falta pedir que se dibujen las gráficas de las distintas funciones. Con la instrucción **DispGraph** del submenú **I/O** dibujaremos todas las funciones salvo aquellas en las que B vale cero, y que dibujaremos con una línea vertical. Así pues, crearemos un bucle para detectar la restricción en la que B es cero: lo habíamos hecho poniendo un 1 en el elemento correspondiente de la matriz C.

```
PROGRAM:PLGEN
:Disp "INTR. F.
OBJET."
:Disp "A,B EN AX
+BY=0"
:Input "A=",A:In
Put "B=",B
:"(-A/B)X"→Yg
```

La instrucción **For**( *variable, principio, final, incremento*), va aumentando de uno en uno dicha variable si no se indica ningún incremento. Se realizarán las instrucciones que sigan hasta que se llegue al valor final de la variable, y se pasará el control a las instrucciones que continúen tras un End.

```
PROGRAM:PLGEN
:DispGraph
:For(I,1,N)
:If [C](1,I)=1
:Line([A](3,I)/[
A](1,I),Ymin,[A]
(3,I)/[A](1,I),Y
max):End
```

La instrucción **Line** (que se encuentra en el menú **DRAW**) tiene como sintaxis: **Line(X1,Y1,X2,Y2)** nos dibujará una línea desde el punto de coordenadas (X1,Y1) al punto (X2,Y2). Con esto, ya deberemos reconocer cual es la región factible de nuestro problema, así como por dónde se puede encontrar la solución.

FnOff	{1,N}→dim([C]
PlotsOff	Fill(0,[C])
AxesOn	Lbl 0
.1→Xmax:.1→Ymax:ClrHome	ClrHome
Input «NUMERO RESTR.?»,N	Disp «RESTRICCION»
ClrHome	Disp «NUMERO»
Disp «INTRODUCE»	Output(2,8,I)
Disp «RESTRICCIONES:»	Input «A=»,A:A→[A](1,I)
Disp «A,B,C PARA»	Input «B=»,B:B→[A](2,I)
Disp «AX+BY<=C»	Input «C=»,C:C→[A](3,I)
Output(6,4,»PULSA ENTER«)	If I=1
Pause	«([A](3,1)-[A](1,1)X)/[A](2,1)»→Y <sub>1</sub>
1→I	If I=2
{3,N}→dim([A]	«([A](3,2)-[A](1,2)X)/[A](2,2)»→Y <sub>2</sub>
Fill(0,[A])	If I=3



```

«([A](3,3)-[A](1,3)X)/[A](2,3)»→Y3
If I=4
«([A](3,4)-[A](1,4)X)/[A](2,4)»→Y4
If I=5
«([A](3,5)-[A](1,5)X)/[A](2,5)»→Y5
If I=6
«([A](3,6)-[A](1,6)X)/[A](2,6)»→Y6
If I=7
«([A](3,7)-[A](1,7)X)/[A](2,7)»→Y7
If I=8
«([A](3,8)-[A](1,8)X)/[A](2,8)»→Y8
If A=0:Then
Goto R
Else
If (3/2)(C/A)>Xmax
(3/2)C/A→Xmax
End
Lbl R
If B=0:Then:1→[C](1,I)
Goto S:Else
If (3/2)(C/B)>Ymax
(3/2)C/B→Ymax
End
Lbl S
(I+1)→ I
If I<=N
Goto 0
-(Xmax/3)→Xmin
-(Ymax/10)→Ymin
int(((Xmax-Xmin)/20)→Xscl
int(((Ymax-Ymin)/20)→Yscl
ClrHome
Disp «INTR. F. OBJET.»
Disp «A,B EN AX+BY=0»
Input «A=»,A:Input «B=»,B
«(-A/B)X»→Y9
DispGraph
For(I,1,N)
If [C](1,I)=1
Line([A](3,I)/[A](1,I),Ymin,[A](3,I)/
[A](1,I),Ymax)
End

```

**Actividad 3.1.-** Se plantea editar una nueva revista de viajes y gastronomía con un total de 150 páginas por número. Se toma la decisión de no destinar a publicidad más de 40 páginas. Cada artículo de viajes se paga a 10.000 ptas. la página y cada artículo de gastronomía a 6.000 ptas. Por otra parte, se cobran 25.000 ptas. por plana de publicidad. El número de páginas dedicadas a viajes ha de ser al menos, igual al dedicado a gastronomía.

¿Cual ha de ser el número de páginas dedicadas a cada sección y cual a publicidad, para que el coste sea mínimo, suponiendo que los gastos de edición sean los mismos en cualquier caso?



**Actividad 3.2.-** En un problema de programación lineal queremos minimizar la función lineal  $3X + 4Y + 2(10-X) + 3(18-Y)$  con las restricciones siguientes:  $X \geq 0$ ,  $Y \geq 0$ ,  $10 - X \geq 0$ ,  $18 - Y \geq 0$ ,  $X + Y \leq 13$ ,  $(10 - X) + (18 - Y) \leq 15$

Haz la representación gráfica del conjunto factible, encuentra las coordenadas de todos sus vértices y encuentra las soluciones óptimas.

**Actividad 3.3.-** Una compañía fabrica estantes para libros. Para cada estante se necesitan tres máquinas. Hay dos modelos de estantes: normal y extra (de mejor calidad). Los estantes normales se fabrican en 5'6 horas: 2'0 horas en máquina I; 1'2 horas en máquina II; 2'4 horas en máquina III

Se usan las mismas máquinas para los estantes extra:

2'0 horas en máquina I; 2'4 horas en máquina II; 0'8 horas en máquina III

Lás máquinas funcionan durante 48 horas (cada una) como máximo, por semana. La ganancia por estante normal es de 192 pesetas, mientras que la ganancia por estante extra es de 384 pesetas. ¿Cuántos estantes de cada clase tendría que producir el fabricante para obtener una ganancia máxima? ¿Hay más de una solución?

**Actividad 3.4.-** Un fabricante desea gastar una parte de su presupuesto haciendo propaganda por radio o televisión. Quiere unos minutos de emisión en la radio (r) y unos minutos en la televisión (t). Pone unas cuantas condiciones adicionales:

- quiere por lo menos 30 segundos de radio;
- quiere por lo menos 3 minutos de televisión;
- la radio cuesta 80.000 pesetas por minuto, mientras que la televisión cuesta 320.000 pesetas por minuto;
- la cantidad máxima a gastar es de 1.600.000 florines;
- quiere por lo menos el doble de tiempo en la televisión que en la radio.

Dibuja la región factible que se obtiene como consecuencia de estas restricciones. Las propagandas emitidas tienen que ser múltiplos de 30 segundos. ¿Cual es la mayor cantidad de minutos que puede emitirse? Con posterioridad el presupuesto se reduce en un 20%. ¿Qué posibilidades existen ahora?





## 4.- CONTRASTE DE HIPÓTESIS

**Actividad 4.1.-** Se lanza 60 veces un dado, con los siguientes resultados:

Resultados obtenidos	1	2	3	4	5	6
Frecuencias	17	7	8	11	8	9

Podremos considerar que el dado es correcto si, desde el punto de vista estadístico, la diferencia entre las frecuencias observadas para cada resultado y las frecuencias esperadas son debidas únicamente al error de muestreo, es decir, al azar, y no son debidas a otras causas. Para verificar esto es conveniente utilizar el valor del estadístico test de la  $\chi^2$ , que mide estas discrepancias:

$$\chi^2 = \sum_{i=1}^s \frac{(X_i - E_i)^2}{E_i}$$

Este valor mide las desviaciones entre los valores observados  $X_i$  y los valores esperados  $E_i$ . Cuando el estadístico es grande, quiere decir que los valores esperados y los observados se distancian, mientras que si es pequeño esta distancia no es grande y los valores obtenidos no discrepan en gran medida de lo esperado. Ahora bien, hay que determinar qué valor se considera como “grande” a la hora de decidir respecto de esta discrepancia.

Podemos imaginar la distribución en el muestreo del estadístico test, es decir, la distribución que seguiría  $\chi^2$  si tuviéramos muchas muestras obtenidas de un dado “normal”. También podemos pensar que los valores más alejados son los más difíciles de obtener, y que hay un valor hasta el cual llegan el 95% de los valores obtenidos. Si el valor del estadístico de nuestra muestra es un caso “raro”, o sea, poco probable de obtener suponiendo que el dado es normal, entonces su valor caería a partir de ese 5% restante de valores, en lo que llamamos la **cola de la distribución**, y nuestra conclusión sería que los datos de la muestra rechazan la hipótesis de considerar como correcto el dado, a un nivel de significación del 5%.

Lo que haremos, pues, será el cálculo del estadístico  $\chi^2$ , y suponiendo que el dado es normal, averiguar la probabilidad de que  $\chi^2$  tome este valor en la cola de la distribución, a lo cual llamaremos **p-valor** de los datos.



En primer lugar, lo que haremos será guardar en una lista los valores obtenidos en la muestra: en  $L_1$  colocaremos todas las frecuencias obtenidas. A continuación, guardaremos en la variable M la frecuencia total (que en este caso es 60) sumando todas las frecuencias de la lista  $L_1$ . En la variable P guardaremos la dimensión de la lista, de manera que  $M/P$  me proporciona la frecuencia esperada de cada resultado. Ahora creamos el programa al que llamaremos **CHI2**.

L1	L2	L3	1
17	-----	-----	
7			
8			
11			
8			
9			
-----			

L1 = {17, 7, 8, 11, 8...}

Con la tecla  $\boxed{2nd}$  [**LIST**] vamos al submenú **MATH** donde aparece la instrucción que me permitirá sumar los elementos de la lista. Asignamos esta suma a la variable M.

```
NAMES OPS MATH
1:min(
2:max(
3:mean(
4:median(
5:sum(
6:Prod(
7:stdDev(
```

La instrucción siguiente **dim(** se obtiene con la tecla  $\boxed{MATRX}$  dentro del submenú **MATH**. Asignamos la dimensión de  $L_1$  a la variable P. Ahora, limpiamos la pantalla por si hubiera alguna información anterior e inicializamos con 0 la variable X.

```
PROGRAM:CHI2
:sum(L1)→M
:dim(L1)→P
:ClrHome
:0→X
:
:
:
:
```

A partir de aquí, y mediante un bucle, acumularemos en la variable X, cada uno de los sumandos del estadístico test. Por último, sólo nos faltará mostrar el valor calculado, que haremos con la instrucción **Disp**. Obtenemos que el valor del estadístico es 6'8. ¿Es normal o es demasiado grande? Una forma de saberlo es acudir a las tablas de la distribución  $\chi^2$ , con 5 grados de libertad, y observamos que el valor 7'29 proporciona un P-valor de 0'20, y puesto que el valor obtenido es menor eso querrá decir que el P-valor de los datos de la muestra es mayor del 20%, luego hemos obtenido una muestra que cae dentro del 80% de las muestras consideradas "normales". Si queremos obtener el P-valor exacto, sólo tenemos que acudir al menú **DISTR** y calcular el valor de la distribución hasta 6'8.

```
PROGRAM:CHI2
:0→X
:For(I,1,P,1)
: X+(L1(I)-(M/P))^
: 2/(M/P)→X
:End
:Disp "VALOR DEL
:EST.:",X
```





La sintaxis de la instrucción es: (*valor mínimo, valor máximo, grados de libertad*). El P-valor exacto sale alrededor de 0'2360. Como es mayor del 0'05 (índice de significación que se considera aceptable), la conclusión sería que los datos no permiten rechazar la hipótesis de considerar el dado correcto.

¿Podemos obtener esta conclusión sin necesidad de acudir a las tablas? Como comentamos al principio, podríamos simular el lanzamiento de un dado normal 60 veces y calcular el estadístico test. Y esto mismo, hacerlo unas 100 veces, con lo cual obtendríamos 100 valores del estadístico test de un dado normal, o sea, una distribución de frecuencias del estadístico test, que se aproximará a su distribución en el muestreo. Haremos un programa que realice dicha simulación, y lo llamaremos **SIMCHI2**.

```
x2cdf(0,6,8,5)
.7640554628
1-Ans
.2359445372
```

Para dar al programa una utilidad más general, permitiremos que el usuario decida el número de resultados elementales del experimento aleatorio, el número de pruebas de cada serie y el número de series, asignando estos valores a los parámetros P, M y N, respectivamente. Guardaremos los valores en dos listas: L<sub>1</sub> y L<sub>2</sub>. En L<sub>1</sub> pondremos los 100 valores del estadístico test obtenidos en cada una de las simulaciones, mientras que en L<sub>2</sub> pondremos las frecuencias absolutas de haber obtenido cada uno de los valores: L<sub>2</sub>(I) contendrá la frecuencia absoluta de valores "I" obtenidos.

La primera acción a hacer será borrar dichas listas, por si hubiera algún otro dato no deseado: lo realizaremos con **ClrList** del submenú **EDIT** en **[STAT]**. Asignamos a la dimensión de la L<sub>1</sub> el valor de la variable N. Ahora iniciamos el bucle de "llenado" de dicha lista, es decir, de obtención de las 100 muestras de 60 tiradas. Así, el bucle irá desde 1 hasta N. Primero dimensionamos la lista L<sub>2</sub> con el valor guardado en P. Después, obtenemos la muestra aleatoria con un nuevo bucle, que irá desde 1 hasta M (que en nuestro caso será 60).



La muestra aleatoria debe consistir de 60 valores comprendidos entre 1 y 6, por lo que usaremos la instrucción **randInt**( que se encuentra en el submenú **PRB** del menú **MATH**). A continuación aumentamos en una unidad la frecuencia absoluta del valor obtenido en la posición correspondiente de la lista  $L_2$ , y cerramos el bucle.

```
PROGRAM:SIMCHI2
ERIM.",N
:ClrList L1,L2
:N→dim(L1)
:For(I,1,N,1)
:P→dim(L2)
:For(J,1,M,1)
:randInt(1,6)→S
```

Sólo nos falta calcular el valor del estadístico test para la muestra obtenida. Para ello, crearemos un nuevo bucle que realice el sumatorio como vimos ya anteriormente. Guardamos el valor obtenido en la posición correspondiente de la lista  $L_1$ . Sólo nos resta borrar la lista  $L_2$  para volver a realizar otra simulación y cerrar el bucle.

```
PROGRAM:SIMCHI2
:L2(S)+1→L2(S)
:End
:0→E
:For(K,1,P,1)
:E+((L2(K)-(M/P)
)²/(M/P)→E
:End
```

Una vez hechos los 100 experimentos, haremos una ordenación de los valores del estadístico test obtenidos en orden decreciente con la instrucción **SortD**(. Ahora, mediante un bucle, compararemos el valor de nuestro estadístico test obtenido de nuestra muestra inicial (que se encuentra en la variable X) con los valores ordenados de la lista, para ver cuántos hay mayores. Cuando encontremos un valor de  $L_1$  que sea menor que X, quiere decir que todos los vistos hasta ahora han sido mayores o iguales, y la posición anterior en la lista que se guarda en la variable L indicará cuántos valores hay mayores que X. Si dividimos L entre N, obtendremos el porcentaje de valores de la lista que son mayores que X, lo cual será una estimación del P-valor de nuestro estadístico test.

```
PROGRAM:SIMCHI2
:E→L1(I)
:ClrList L2
:End
:SortD(L1)
:For(I,1,N,1)
:If X≤L1(I)
:I→L
```

```
PROGRAM:SIMCHI2
:For(I,1,N,1)
:If X≤L1(I)
:I→L
:End
:L/N→L
:Disp "P(X²≥E)="
,L
```



Podemos ahora hacer la simulación con nuestra calculadora ejecutando el programa.

Programa para el cálculo del estadístico  $\chi^2$

```
:sum (L1, → M
:dim (L1 → P
:ClrHome
:For(I,1,P,1)
:X+(L1(I)-(M/P))^2/(M/P) → X
:End
:Disp "VALOR DEL EST.:", X
```

```
:N → dim (L1
:For(I,1,N,1)
:P → dim (L2
:For(J,1,M,1)
:(1+int (Prand)) → S
:L2(S)+1 → L2(S)
:End
:0 → E
:For(K,1,P,1)
:E+((L2(K)-(M/P))^2/(M/P)) → E
:End
:E → L1(I)
:ClrList L2
:End
:SortD(L1)
:For(I,1,N,1)
:If X ≤ L1(I)
:I → L
:End
:L/N → L
:Disp «P(X2 ≥ E)», L
```

Programa para calcular mediante simulación el p-valor en un test  $\chi^2$

```
:Input «NUM. POSIBILIDADES», P
:Input «TAM. MUESTRA », M
:Input «NUM. EXPERIM. », N
:ClrList L1, L2, L3, L4, L5, L6
```

**Actividad 4.2.-** Una tabla de 1.000 números aleatorios se puede escribir de la siguiente forma:

Cifra	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
Frecuencia	122	90	92	120	90	92	110	100	76	108

Probar la calidad de la tabla, con ayuda de  $\chi^2$ .



## 5.- BIBLIOGRAFÍA

AZARQUIEL. (1985). “*Curso inicial de Estadística en el Bachillerato*”. (I.C.E. de la U. Autónoma: Madrid)

BATANERO, M.C. DIAZ, J. Y ESTEPA, A. (1991). “*Curso de Estadística Aplicada basado en el uso de ordenadores*”. (Los autores: Granada)

CERO, GRUPO.(1983). “*Análisis y Estadística (Matemáticas 3º de B.U.P.)*”.(I.C.E. Universidad de Valencia)

DIAZ, J. BATANERO, M.C. Y ESTEPA, A. (1988). “*Laboratorio de Estadística*”. (Los autores: Jaén)

ESTEPA A. (1990). “*Enseñanza de la Estadística basada en el uso de Ordenadores: Un estudio exploratorio*”.(Memoria de tercer ciclo: U. de Granada)

SAMUELS, M.L. (1989) “*Statistics for the life sciences*”. (San Francisco, CA: Dellen Publishing)

SARRAMONA, J. (1980). “*Investigación y Estadística aplicadas a la educación*”. (Ediciones CEAC, Barcelona)

TEXAS INSTRUMENTS

