

Ejercicio

Se envía a través de una conexión RS232 8N1 de 9600 baudios una trama de datos que debe ser recibida por una PC.

La trama tiene el siguiente formato :

*	N	N	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Donde:

- ‘*’ es el carácter asterisco (2Ah) que representa el comienzo de trama.
- X son 20 caracteres cualquiera de información.
- N son 2 caracteres (1 Word) que representan la sumatoria de los 20 caracteres de información.

Se deberá realizar un programa en Assembler que:

- Mediante polling espere y lea los caracteres que se reciben por el COM1 (3F8h) y al recibir un asterisco comience la lectura de la trama. La información será guardada en un vector de 20 caracteres.
- Además se deberá utilizar una variables mas a la cual se le asignarán los siguientes valores:
0: trama recibida correctamente.
1: no coincide la suma de los 20 bytes con la sumatoria recibida en los 2 primeros bytes.

Nota:

No es necesario inicializar el puerto, solo se debe comenzar a leer.

El carácter recibido es leído en el COM1, dirección 3F8h.

```
PROGRAMA DE AYUDA PARA LEER EL PUERTO COM1

no_hay_dato:
    mov     dx,COM1+5      ; (Reg LSR)
    in     al,dx          ; si DR=1 dato a leer
    and    al,1           ; si DR=0 buffer vacío
    jz     no_hay_dato
    mov    dx,COM1
    in    al,dx
```

Ejercicio

Se envía a través de una conexión RS232 8N1 de 9600 baudios una trama de datos que debe ser recibida por una PC a través del COM1 y reenviada a otros dispositivos conectados a los demás puertos serie.

La trama tiene el siguiente formato:

*	N	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Donde:

- ‘*’ es el caracter asterisco (2Ah) que representa el comienzo de trama.
- X son 20 caracteres cualquiera de información.
- N es un byte que represente al puerto COM por el que se debe enviar la cadena recibida.

Se deberá realizar un programa en Assembler que:

- Mediante polling espere y lea los caracteres que se reciben por el COM1 (3F8h) y al recibir un asterisco comience la lectura de la trama. El segundo byte recibido será el puerto COM por el que se debe reenviar la trama, este número puede ser 2, 3 o 4.
- La cadena reenviada comenzará también con un asterisco debiendo ser de la siguiente forma:

*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Nota:

No es necesario inicializar los puertos, solo se debe comenzar a leer.

El carácter recibido es leído en el COM1.

RUTINAS DE AYUDA PARA LEER Y RECIBIR DATOS			
COM1	EQU	3f8h	
COM2	EQU	2f8h	
COM3	EQU	3e8h	
COM4	EQU	2e8h	
Recibir Dato			
no_hay_dato:			
mov	dx,COM1+5		; (Reg LSR)
in	al,dx		; si DR=1 dato a leer
and	al,1		; si DR=0 buffer vacío
jz	no_hay_dato		
mov	dx,COM1		
in	al,dx		
Enviar Dato			
mov	dx,COMn		
mov	al,b1		
out	dx,al		

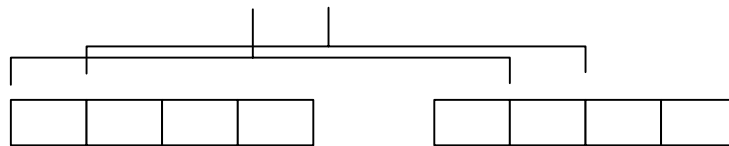
Ejercicio

En un sistema de microprocesador y ADC de 12 bits es necesario realizar un promedio antes de procesar los datos obtenidos del ADC.

Se deberá realizar un programa en Assembler que:

- lea por polling el ADC.
- promedie por el método de “moving average” utilizando 20 tomas.
- devuelva el resultado en AX.

Antes de que se completen las 20 tomas, el software devolverá el promedio de las n tomas capturadas hasta ese momento.



PROGRAMA DE AYUDA PARA LEER EL ADC

```
no_hay_dato:
    mov    dx,202          ; (Reg LSR)
    in     al,dx           ; si 202h=1 dato a leer
    and    al,1           ; si 202h=0 buffer vacío
    jz     no_hay_dato
    mov    dx,200
    in     ax,dx
```

Lec 1 Lec 2

T1 T2 T3 T4 ...

Ejercicio

Se debe realizar un programa en Assembler que calcule los primeros 100 números primos por el método de la “Criba de Eratóstenes”.

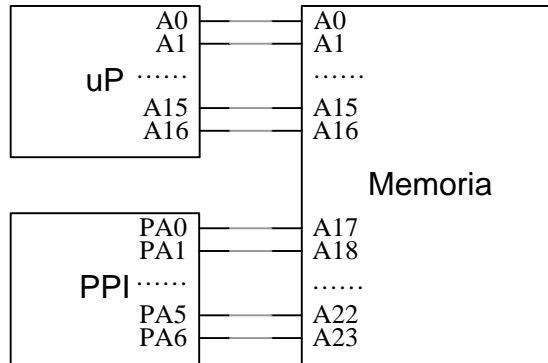
El método se resume en estos pasos:

- cargamos un arreglo de números naturales desde 2 hasta 100.
- comenzamos por el número 2, lo dejamos, pero a partir de él eliminamos los números que sean múltiplos de 2.
- el primer número de los que quedan es el 3, lo dejamos y desde el número 3 eliminamos los números que sean múltiplos de 3.
- el siguiente número de los que quedan es el 5, lo dejamos y desde el número 5 eliminamos los números que sean múltiplos de 5.
- así vamos avanzando, cuando llegamos a un número que no ha sido eliminado lo dejamos, pero a partir de él eliminamos los números que sean múltiplos de él. Así hasta el final.

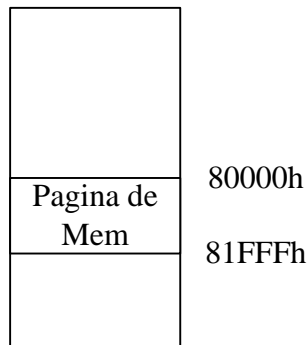
Finalmente habrán quedado solamente números primos.

Ejercicio

En un sistema de microprocesador, es necesario leer una memoria de 16 Mb, para lo cual se dispondrá del puerto de una PPI para completar el direccionamiento de la memoria



La memoria está dispuesta desde la dirección 80000h hasta la 81FFFh (128 Kb)



Se deberá realizar un programa en Assembler que:

- reciba en DX AX la posición a leer (24 bits)
- extraiga de esa posición los 7 bits a enviar al puerto de la PPI, que corresponde a los 7 bits mas significativos de la memoria
- calcular el segmento y puntero.
- leer el dato y enviarlo a AL

Nota: la PPI se encuentra en la dirección 200h, y ya se encuentra configurada, esto significa que para escribir al puerto, se deberá utilizar:

```
mov    dx,200
out    al,dx
```