

ALUMNO: _____

1. Explique qué registros forman parte del registro base Y. Utilice las instrucciones necesarias para que Y apunte a la primera dirección de la memoria RAM. **(0,5 pts)**

Los registros R29 y R28 son los que conforman Y. R 29 es el byte superior o más significativo (YH) y R28 el byte inferior o menos significativo (YL).

La primera dirección de la memoria RAM se encuentra en la dirección \$100 de la memoria de datos, por tanto, debemos de escribir la dirección \$100 en Y. Podemos hacerlo con la instrucción LDI, de diferentes formas:

LDI R29, 1	LDI YH,1	LDI YH,0x01
LDI R28,0	LDI YL,0	LDI YL,0X00

2. En lenguaje ensamblador podemos encontrar directivas de ensamblado e instrucciones ejecutables. Explique la diferencia entre ambas categorías. **(0,5 pts)**

Las directivas de ensamblado no generan código máquina, son una ayuda para facilitar la programación y la generación del código máquina.

Las instrucciones ejecutables pertenecen al juego de instrucciones del procesador y generan código máquina.

3. ¿Cuántos registros de E/S tiene el microcontrolador Atmega328P? ¿Y de E/S extendida? Mencione un registro de cada uno de dichos grupos. **(1 pto)**

60 registros de E/S y 160 de E/S extendida

4. Suponga R20 = \$25 y R21=\$FE indique el valor final de los registros y de los bits de estado Z, C, N, V y S tras ejecutar la instrucción ADD R20, R21. Razone qué indican dichos valores (los que usted ha indicado) para este caso particular. **(1 pto)**

Al sumar las palabras \$25 Y \$FE tendremos la suma de los binarios 0010 0101 y 1111 1110, esto producirá el resultado 0010 0011 con Z=0 (el resultado no es 0), C=1 (desborda si se consideran números sin signo pues tendríamos 37+254), N = 0 (el bit en la posición más a la izquierda del resultado es 0), V = 0 (no desborda si se consideran números con signo pues tendríamos 37-2) y S = 0 (el signo verdadero o XOR(S,V) es positivo).

5. Usando las instrucciones LDS y STS sustituya cada una de las siguientes instrucciones, si esto es posible, de modo que se consiga el mismo resultado: **(1 pto)**

MOV R16,R0	LDS R16,0 o bien STS 16,R0
IN R16,PINB	LDS R16,\$23
LDI R16,16	no se puede

6. Escriba las líneas de programa necesarias para configurar los pines del puerto B como entradas y los pines PD2 y PD6 como salida. Tenga en cuenta que ha de proporcionar resistencia de pull-up para todas las entradas. **(1 pto)**

```
LDI R16,0
OUT DDRB,R16
LDI R16,0xFF
OUT PORTB,R16
SBI DDRD,2
SBI DDRD,6
```

7. Suponga que ya se ha configurado el pin PD2 como salida y los pines del puerto B como entradas. Escriba las líneas de programa necesarias para leer el puerto B y en caso de que la lectura sea el dato \$80 escribir un 1 en la salida (en el pin PD2). **(1 pto)**

```
IN R16,PINB
CPI R16,$80
BRNE SIGUE
SBI PORTD,2
SIGUE: ...
```

8. Escriba las líneas de programa necesarias para realizar las siguientes tareas: **(1 pto)**

- a. Escribir un 1 en el bit más significativo de R10.

```
LDI R16,0b1000000
OR R10,R16
```

- b. Escribir un 1 en el bit más significativo del registro nº 12 de E/S.

```
SBI 12,7
```

- c. Escribir el dato 2000 en la posición de memoria 0x2000

Lo hacemos en notación little endian, escribiendo en primer lugar el byte menos significativo de 2000.

```
LDI R16,low(2000)
STS 0x2000,R16
LDI R16,high(2000)
STS 0x2001,R16
```

- d. Incrementar el valor de X en 10 unidades

```
ADIW X,10 o ADIW R26,10
```

9. Desarrolle un programa que permita obtener una señal cuadrada de frecuencia 5Hz conmutando el pin PC2 del microcontrolador. Para ello se programará el temporizador para que provoque una interrupción cada 0,1 segundos. La rutina de interrupción será la encargada de conmutar el valor de PC2. No olvide:
- Escribir las líneas necesarias de la tabla de vectores de interrupción.
 - Configurar el puerto PC2.
 - Configurar el temporizador para que genere las interrupciones al ritmo requerido. (considere que el microcontrolador trabaja a 1MHz).
 - Escribir la rutina de interrupción.

(3 ptos)

A frecuencia 1Mhz: → en 1s pasarían 1.000.000 de ciclos

→ en 0,1s pasarían 100.000 ciclos

Si dividimos la frecuencia por 8 (clk/8)

→ en 0,1s pasarían $100.000/8$ ciclos = 12500 ciclos

Haremos OCR1A ← 15625, seleccionaremos modo CTC (TCCR1B3 = 1) y frecuencia ck/8 (TCCR1B2.0= 0b010)

Activaremos las interrupciones haciendo: I=1 y TIMSK1= 1

```
.include "m328pdef.inc"
```

```
.cseg
```

```
.org 0
```

```
jmp comienzo
```

```
.org 0x16
```

```
jmp IntTimer1
```

```
.org 0x30
```

```
.def tmp =r16
```

```
comienzo:      sbi ddrc,2                ;configura PC2 como salida
```

```
                clear tmp
```

```
                sts tcnt1h,tmp
```

```
                sts tcnt1l,tmp
```

```
                ldi tmp,high(12500)        ;ocr1a = 12500
```

```
                sts ocr1ah,tmp
```

```
                ldi tmp,low(12500)
```

```
                sts ocr1al,tmp
```

```

        ldi    tmp,2
        sts    tmsk1,tmp    ;habilita interrupción CTC
        sei    ;habilita interrupciones

        ldi    tmp, 0b00001010    ;f = clk/8 y modo CTC
        sts    tccr1b,tmp

        fin:   rjmp  fin

IntTimer1:  push tmp
            in  tmp,sreg

            in  r20,pinc
            com r20
            out portc,r20

            out tmp,sreg
            pop tmp
            reti

```

también podría ser:

```

IntTimer1:  push tmp
            in  tmp,sreg

            sbic portc,2
            rjmp poner0
            rjmp poner1

poner0:    cbi portc,2
            rjmp finrutina

poner1:    sbi portc,2

finrutina:  out tmp,sreg
            pop tmp
            reti

```