

PARCIAL 1

1. El siguiente programa envía un parámetro por la pila a SUBROUT. ¿Qué valor queda almacenado en BH? (10 ptos)

```
ORG 1100h
MSG DB "Hola"
ORG 2000h
MOV AX, OFFSET MSG
PUSH AX
CALL SUBROUT
...
```

```
ORG 3000h
SUBROUT: MOV BX, SP
ADD BX, 1
MOV BH, [BX]
...
```

Respuesta:

BH = 11h ✓

Ejercicio Puntos

1	10
2	20
3	35
4	25
Total	90

Nota: Recuerde que en VonSim los valores de 2 bytes se guardan con la parte menos significativa en la dirección más chica y viceversa.

2. El siguiente programa para VonSim lee caracteres por teclado y los envía a la impresora a través del HANDSHAKE a medida que se van leyendo. El programa termina cuando se lee el carácter "z" o se presiona F10. Completar las instrucciones faltantes. (20 ptos)

```
DATQ EQU 40h
ESTADO EQU 41h
IMR EQU 21h
INT0 EQU 24h

org 1000h
car db ?
flag_fin db 0

org 20
dirf10 dw F10

org 3000h
F10: mov flag_fin, 1
mov al, 20h
out 20h, al
iret
```

```
org 3100h
imp: in al, ESTADO
and al, 1
jnz imp
mov al, car ✓
out DATQ, al ✓
ret
```

```
org 2000h
cli
mov al, 11111110b
out IMR, al ✓
mov al, 5 ✓
out INT0, al
```

```
in al, ESTADO
and al, 01111111b
out ESTADO, al
sti
mov bx, offset car
loop: cmp flag_fin, 0
jnz fin
int 6
cmp car, 'z'
jz fin
call imp
jmp loop
fin: INT 0
end
```

3. Escribir un programa para VonSim que deberá utilizar las luces y llaves de la siguiente forma: (40 ptos)

- Cada vez que las llaves cambien de valor, se muestra en pantalla el código ASCII correspondiente a los 8 bits de las llaves. Por ejemplo, si todas las llaves, salvo la del bit 0 y 6, están apagadas, se muestra el carácter 'A' correspondiente al código ASCII "01000001".
- Cada vez que se encuentre prendida la última llave (la del bit más significativo), se deben invertir las luces (las que estaban prendidas se apagan y viceversa).
- En el caso particular en que todas las llaves estén apagadas, se deben apagar todas las luces y finalizar el programa.

Las funciones "a", "b" y "c" deben implementarse utilizando subrutinas.

4. Escribir un programa para VonSim que envíe la cadena de caracteres "Arquitectura de Computadoras" a un dispositivo nuevo, conectado a los 8 bits del puerto PA. Este dispositivo recibe la cadena de a un carácter a la vez. Para que el dispositivo reconozca que se va a enviar un dato, luego de enviar un carácter debe enviar el valor 255. El programa debe finalizar cuando se han enviado todos los caracteres de la cadena, o cuando pasan 30 segundos desde que inició el programa, lo que suceda primero. (30 ptos)

Ejemplo para enviar la cadena "ABCD": Envío de "A" → Envío de 255 → Envío de "B" → Envío de 255 → Envío de "C" → Envío de 255 → Envío de "D" → Envío de 255

Nota: para comunicarse con el dispositivo no es necesario realizar una consulta de estado ni hacerlo mediante interrupciones, el protocolo solo requiere que envíe un 255 y un carácter de forma alternada.

PARCIAL 2

1. El siguiente programa envía un parámetro por la pila a SUBROUT. ¿Qué valor queda almacenado en DL? (10 ptos)

```

1000H  ORG 1000h
1001  CAD DB "Hola"
1002  ORG 2000H
1003  MOV DX, OFFSET CAD
      PUSH DX
      CALL SUBROUT
      ...

```

```

      ORG 3000H
SUBROUT: MOV BX, SP
          ADD BX, 3
          MOV DL, [BX]

```

Diagrama de memoria:

1000	7FFD	12	2-SP	...
1001	7FFE	13		
1002	7FFF	00		
1003	8000	10		[BX]

Respuesta:

DL = "H" MAYÚSCULA
(Hexadecimal)

Ejercicio	Puntos
1	
2	
3	
4	
Total	

Nota: Recuerde que en VonSim los valores de 2 bytes se guardan con la parte menos significativa en la dirección más chica y viceversa.

2. El siguiente programa para VonSim lee caracteres por teclado y los envía a la impresora a través del PIO a medida que se van leyendo. El programa termina cuando se lee el carácter "." Completar las instrucciones faltantes. (20 ptos)

```

PA EQU 30h
PB EQU 31h
CA EQU 32h
CB EQU 33h

```

```

org 1000h
car db ?

```

```

org 2000h
mov al, 0
out CB, al
mov AL, 1111101b
out CA, al
in al, PA
and al, 11111101b
out PA, al
mov bx, offset car
loop: int 6
      cmp bx, "."
      jz fin
      call imp
      jmp loop
fin: INT 0
end

```

```

imp: in al, PA
      AND AL, 1
      jnz imp
      mov al, car
      out PB, al
      in al, PA
      OR AL, 0000010b
      out PA, al
      in al, PA
      and al, 11111101b
      out PA, al
      ret

```

3. Escribir un programa para VonSim que deberá utilizar las luces y llaves de la siguiente forma: (40 ptos)

- Cada vez que las llaves cambien de valor, se actualizan las luces a su estado opuesto. De modo que si las llaves están en el estado "00011010" las luces tendrán el estado "11100101".
- Cada vez que encuentre la primera llave (la del bit menos significativo) prendida, mostrar en pantalla el mensaje "Arquitectura de Computadoras".
- En el caso particular en que todas las llaves estén apagadas, mostrar en pantalla el mensaje "Fin de programa" y finalizar el mismo.

Las funciones "a", "b" y "c" deben implementarse utilizando subrutinas.

4. Escribir un programa para VonSim que envíe la cadena de caracteres "Universidad Nacional de La Plata" a un dispositivo nuevo, conectado a los 8 bits del puerto PA. Este dispositivo recibe la cadena de a un carácter a la vez. Para que el dispositivo reconozca que se va a enviar un dato, antes de enviar un carácter debe enviar el valor 0. El programa debe finalizar cuando se han enviado todos los caracteres de la cadena, o cuando se presiona la tecla F10, cancelando el envío cancelando el envío de los caracteres que restan. (30 ptos)

Ejemplo para enviar la cadena "ASDF": Envío de 0 → Envío de la "A" → Envío de 0 → Envío de la "S" → Envío de 0 → Envío de la "D" → Envío de 0 → Envío de la "F"

Nota: para comunicarse con el dispositivo no es necesario realizar una consulta de estado ni hacerlo mediante interrupciones, el protocolo solo requiere que envíe un 0 y un carácter de forma alternada.

nte programa envía un parámetro por la pila a SUBRUT. ¿Qué valor queda almacenado en CL?

```
250h
AD DB "Hola"
2000H
MOV AX, OFFSET CAD
PUSH AX
CALL SUBRUT
...
```

```
ORG 3000H
SUBRUT: MOV BX, SP
        ADD BX, 2
        MOV CL, [BX]
        ... 316
```

Respuesta:

CL = 50H X

Ejerc

1

2

3

4

Total

PARCIAL 3

recuerde que en VonSim los valores de 2 bytes se guardan con la parte menos significativa en la dirección viceversa.

siguiente programa para VonSim lee caracteres por teclado y los envía a la impresora a través del PIO a se se van leyendo. El programa termina cuando se lee el carácter "." Completar las instrucciones faltantes. (2

```
PA EQU 30h
PB EQU 31h
CA EQU 32h
CB EQU 33h
```

```
org 1000h
car db ?
```

```
org 2000h
mov al, 0
out CB, al
mov al, 1111101b
out CA, al
mov bx, offset car
loop: int 6
      cmp car, "."
      jz fin
      call imp
      jmp loop
fin: INT 0
end
```

```
org 3500h
imp: in al, pa
     and al, 1
     jnz imp
     mov AL, [CBX]
     OUT PB, AL
     in al, pa
     mov OR AL, 2
     out pa, al
     in al, pa
     or al, 10b
     out pa, al
     ret
```

3. Escribir un programa para VonSim que deberá utilizar las luces y llaves de la siguiente forma: (40 pto)

- Cada vez que las llaves cambien de valor, se actualizan las luces, de modo que si las llaves están en el estado "00011010" las luces tendrán el estado opuesto, es decir, "111001010".
- Cada vez que encuentre la 8va llave (la del bit más significativo) prendida, mostrar en pantalla el mensaje "Buen día".
- En el caso particular en que todas las llaves estén apagadas, mostrar en pantalla el mensaje "Adiós" y finalizar el mismo.

Las funciones "a", "b" y "c" deben implementarse utilizando subrutinas.

4. Escribir un programa para VonSim que envíe la cadena de caracteres "Universidad Nacional de La Plata" a un dispositivo nuevo, conectado a los 8 bits del puerto PB. Este dispositivo recibe la cadena de a un carácter a la vez. Para que el dispositivo reconozca que se va a enviar un dato, antes de enviar un carácter debe enviar el valor 0. El programa debe finalizar cuando se han enviado todos los caracteres de la cadena, o cuando se presiona la tecla F10, cancelando el envío de los caracteres que restan. (30 pto)

Ejemplo para enviar la cadena "ASDF": Envío de 0 → Envío de la "A" → Envío de 0 → Envío de la "S" → Envío de 0 → Envío de la "D" → Envío de 0 → Envío de la "F"

Nota: para comunicarse con el dispositivo no es necesario realizar una consulta de estado ni hacerlo mediante interrupciones, el protocolo solo requiere que envíe un 0 y un carácter de forma alternada.

PARCIAL 4

Arquitectura de Computadoras

Parcial 1 - Fecha 1

30/09/2022

Nombre: [REDACTED]

Apellido: [REDACTED]

DNI: [REDACTED]

Legajo: [REDACTED]

1) ¿Qué posibles instrucciones deben ir en las líneas 10 y 12 del programa para enviar la variable OSO por referencia a la subrutina POTENCIA?

- Si su DNI termina en un dígito impar, utilizar pasaje por registro.
- Si su DNI termina en un dígito par, utilizar pasaje por pila.

10. MOV BX, OFFSET OSO
12. MOV AL, [BX]

```
1  org 1000h
2  oso: db 15
3
4  org 3000h
5  potencia: ...
6          ...
7          ret
8
9  org 2000h
10 ;instrucciones
11 call potencia
12 ;instrucciones
13 int 0
14 end
```

2. Se debe responder a una interrupción del TIMER (solo el TIMER). Se usará el índice o identificador de interrupción (también conocido como posición del vector de interrupciones) 10. ¿Qué registros del PIC deben modificarse para configurarlo y qué valores se le deben asignar?

(24) IMR = 1111101b (25) INT 1 = 10

3. Si el registro CONT vale 0, el registro COMP vale 1, y el programa no cambia estos registros con un OUT, ¿Habrá un pedido de interrupción del timer? Si es así, ¿cuándo se solicitará?

Si habrá y se solicitará en un segundo

4. Escribir un programa para MSX88 que apague el led de la posición X y luego termine; el resto deben quedar prendidos. X en este caso es el último dígito de tu DNI, si está entre 0 y 7 inclusive. Si el último dígito de tu DNI es 8, apagar el 0, y si es 9, apagar el 1. Nota: Los leds tienen posiciones de 0 al 7, de derecha a izquierda (76543210)

```
MOV AL, 0
OUT CB, AL
IN AL, PB
AND AL, 1111101b
OUT PB, AL
```

5. El siguiente programa verifica el estado de las llaves. Si están prendidas la primera y la última llave, y el resto están apagadas, se debe mostrar en pantalla el mensaje "ACTIVADO"; de lo contrario no se debe mostrar nada. Complete las líneas necesarias para que el programa se ejecute correctamente.

```
PIO EQU 30H
ORG 1000H
MSJ DB "ACTIVADO"
FIN DB ?
```

```
ORG 2000H
MOV AL, 0FFh
OUT PIO+2, AL
```

```
IN AL, PIO
CMP AL, 10000001b
JNZ FIN
MOV BX, OFFSET MSJ
MOV AL, OFFSET FIN - OFFSET MSJ
INT 7
FIN: HLT
END
```

6. Escriba un programa que dada una cadena de caracteres almacenada en la memoria, imprima a través de la impresora solamente sus caracteres mayúscula, en el orden en que aparecen. Para ello, deberá implementar y utilizar una subrutina SELECCIONADO que reciba un carácter como parámetro y devuelva verdadero (1) si el carácter era mayúscula y falso (0) de lo contrario. El programa debe esperar a que se hayan enviado todos los caracteres a imprimir para finalizar.

Según el último dígito de su DNI, la impresión debe realizarse (indicar su caso):

- ☐ Dígitos 0 a 3: Por PIO y consulta de estado, con el protocolo usual.
- ☐ Dígitos 4 a 6: Por HANDSHAKE y consulta de estado, con el protocolo usual.
- ☒ Dígitos 7 a 9: Por HANDSHAKE e interrupciones, con el protocolo usual.

```
ORG 1000h
cadena db "T)Elzxf 213!$f%!.oiasdaR.%!RI5 C:d%i...0:.I:Asld6asdS"
fin db ?
```

```
HAND EQU 40h
PIC EQU 20h
IPHAND EQU 10
```

```
ORG 40
dw RUTHAND
```

```
ORG 2000h
CLI
IN AL, HAND+1
OR AL, 00000000b
OUT HAND+1, AL
MOV AL, 11111011b
OUT PIC+1, AL
MOV AL, IPHAND
OUT PIC+6, AL
MOV BX, OFFSET cadena
MOV CL, OFFSET FIN - OFFSET CADENA
STI
LAZO: CMP CL, 0
JNZ LAZO
INT 0
END
```

```
ORG 3000h
SELECCIONADO: MOV DL, 0
CMP BYTE PTR [BX], 41h
JS FINAL
CMP BYTE PTR [BX], 5Bh
JNS FINAL
MOV DL, 1
```

FINAL: RET

RUTHAND: CALL SELECCIONADO

```
CMP DL, 1
JNZ NOIMP
MOV AL, [BX]
OUT HAND, AL
NOIMP: INC BX
DEC CL
MOV AL, PIC
OUT AC, AL
IRET
```


PARCIAL 5

TEMA 1

Parcial 1 - 27/09/2019

Arquitectura de Computadoras 2019

Apellido y Nombre: _____

Nro. de Legajo: _____

Observaciones: Escribir las respuestas con tinta. Cada ejercicio indica su valor en puntos. SE APRUEBA CON 10 PUNTOS. NOTA MAXIMA: 20.

1. ¿Cómo haría para intercambiar los contenidos de los registros AX y BX sin utilizar otros registros? Por ejemplo, si los contenidos iniciales fueran (AX)=1, (BX)=2, el resultado final debería ser (AX)=2, (BX)=1. Escribir las instrucciones para hacerlo en el recuadro. (1 pto)

Push Ax
Push Bx
Pop Bx
Pop Ax

No se intercamian?

2. Si en el PIC del MSX88 el registro IMR = CBH, ¿qué se está indicando? (1 pto)

Si el registro IMR = CBH indica que está activando el handshake, mientras ignoremos los demas...

3. Si el registro SP = 7FF9H antes de completarse una instrucción IRET ¿Qué valor tendrá SP una vez completada la IRET? (1 pto)
SP = 7FF8H

4. Si los puertos de control y de datos de la impresora están conectados a los puertos PA y PB del PIO, respectivamente, y la impresora no está ocupada, ¿qué dos cosas (y dónde) deben hacerse para imprimir un carácter? (1 pto)

Primero revisar si son datos de entrada o salida en CA y CB y luego compararlo con el handshake si son por interrupcion o no para mandar el pulso

5. El siguiente programa lee continuamente caracteres X ASCII de teclado. Si el carácter ASCII X representa alguno de los dígitos '0', '1', '2', ..., '7', entonces se prende el LED X de las luces y se apaga el resto. Es decir, si se lee el dígito '2', debe prenderse solo el LED 2 y el resto debe estar apagado. En el caso de que se lea un ASCII que no represente un dígito del 0 al 7, se deberá informar que el carácter es inválido y leer otro. Completar el programa con las instrucciones faltantes para que funcione correctamente. (1 pto c/instrucción)

```
PB EQU 31H
CB EQU 33H

ORG 1000H
MSJ DB "INGRESAR LED A PRENDER (0-7):"
FIN DB ?
MSJ_ERROR DB "CARÁCTER FUERA DE RANGO"
FIN_ERROR DB ?
LED DB ?
```

```
ORG 2000H
MOV AL, 0
```

out CB, AL

```
LOOP: MOV BX, OFFSET MSJ
MOV AL, OFFSET FIN-OFFSET MSJ
INT 7
```

```
MOV BX, OFFSET LED
```

Int 6

```
MOV CL, LED
SUB CH, 30H
```

```
MOV AH, 0
CMP CL, 8
JNS YERRO
CMP AH, CL
JS LED_ON
JZ LED_ON
```

```
YERRO: mov Bx, offset MSJ-Error
MOV AL, OFFSET FIN_ERROR-OFFSET MSJ_ERROR
INT 7
JMP LOOP
```

```
LED_ON: MOV CH, 1
MULT2: CMP CL, 0
JZ SEND_LED
ADD CH, CH
DEC CL
JMP MULT2
```

SEND_LED: _____

```
FIN_LOOP: JMP LOOP
END
```

6. Complete el siguiente código para configurar el HANDSHAKE de manera que funcione mediante interrupciones (1 pto)

```
HAND_DATO EQU 40H
HAND_ESTADO EQU 41H
```

in AL, Hand_Estado

or AL, 01111111b

OUT HAND_ESTADO, AL

OR 10000000b

7. Escribir un programa para MSX88 que envíe una serie de códigos ASCII hacia un dispositivo externo, a través del puerto PA del PIO. Los datos, que están almacenados en la memoria del procesador a partir de la celda CODIGO, se deben enviar al puerto PA.

forma sincronizada utilizando la interrupción del TIMER, de tal manera que se transmita un caracter ASCII por segundo. El programa debe finalizar una vez transferidos todos los caracteres. (10 pts).

TIMER EQU 10H
PIC EQU 20H
PIO EQU 30H

ORG 1000H
CODIGO DB "DJHOSIFUOJNGSLJK34857783479DFIGHDFKJjcviehfkbgfgi38494KF94R4OFJLSPÑWE3434"
FIN DB ?

org 2000H
Mov AL, 0FDH
out Pict1, AL
cli
Mov EB, offset codigo
Mov DI, offset Fin
int 7
Mov AL, 10
out Pict2, AL
Mov AL, 0
out Timer1, AL
out Pict3, AL

org 40
IP_INT
subtimer
Push AV
Push BV
subtimer

org 3000H
Push AX
Mov AL, 1
out Timer, AL
Int 7
ADD CBXJ
Dec DI
Mov AL, 0
out Timer, AL
cmp DI, 0
JNZ subtimer
Pop AX
Iret

org 2000H
cli
Mov AL, 0FDH
out Pict1, AL
Mov AL, 0
out Pict2, AL
Mov AL, 1
out Timer1, AL
Mov AL, 10
out Pict3, AL
Mov EBX, offset codigo
Mov DI, offset Fin
subtimer
int 7
Int 0

loop impr
con resp
proto co

MAIN: 315
PIO: 2
PIC: 2
INT: 0