

1. El Caché y la Memoria Principal (RAM) son los elementos más veloces de la computadora?

R/ Verdadero

2. Empareje cada Lenguaje de Programación con su Generación respectiva

Segunda: Lenguaje Ensamblador

Quinta: Lenguaje orientado a Inteligencia Artificial

Sexta: Lenguaje Orientado a la computación Cuántica

Cuarta: Lenguaje de Alto Nivel Parametrizable

Tercera: Lenguaje de Alto Nivel Funcional

Primera: Lenguaje de Máquina

3. Empareje cada función con el Bus al que corresponda

Bus de Alimentación: Suministros de voltajes a los dispositivos

Bus de datos: Transmisión de datos entre los dispositivos internos

Bus de control: Selección de emisor y receptor en transacciones del bus

Bus de direcciones: Indicación de la posición de memoria del dato que se desea acceder

4. Los registros de Control y Estado son manipulables por el procesador y algunos visibles al usuario.

R/ Verdadero

5. Analice el siguiente programa en Lenguaje Ensamblador y elija la opción del resultado que se mostrara en pantalla.

.model small

.code

programa:

mov AX, @DATA ; Data se mueve a ax, 0

MOV DS, AX ; Se mueve a DS lo de ax

MOV AX, 00H ; Ax se vuelve 0

MOV BX, 00H ; Bx se vuelve 0

MOV AL, 50H ; AL tiene 01010000

MOV DL, AL ; Mover AL a DL

```
MOV AH, 02 ; Muestra el carácter en DL "p"
INT 21H ; Ejecuta las interrupciones
MOV AH, 00H ; Interrupción = Termina el programa
MOV AL, 03H ;
INT 10H ;

END
```

Respuesta: Se mostrará un error de ensamblaje o enlazado

6. Empareje cada especificación con el Registro correspondiente

Memory Data Register: Contener lo que el CPU o escribe en la memoria, desde un puerto I/O o hacia un puerto I/O, conectado al Bus de datos

Memory Address Register: Contener la dirección del dato que se requiere leer o escribir, conectado al Bus de direcciones.

Instruction Pointer: Contener la posición en memoria de la próxima instrucción a ejecutar

Instruction Register: Contener la instrucción que se está ejecutando, registro de la UC.

7. La arquitectura Harvard se le conoce como la Arquitectura Clásica de Computador. R/ Falso

8. Cada segmento definido es de 64 bytes de memoria. R/ Falso

9. Analice el siguiente programa en Lenguaje Ensamblador y elija la opción del resultado que se mostrara en pantalla.

```
.model small
.code
programa:
    mov AX, @DATA
    MOV DS, AX
    MOV AX, 00H
    MOV BX, 00H
    MOV AL, 40H
```

```
MOV AH, 4CH  
INT 21H  
.  
stack  
END
```

Respuesta: No se mostrará nada en pantalla, porque nunca se imprime.

10. Según la Clasificación por su Abstracción. ¿Qué lenguajes de programación existe?

- Lenguaje de Medio Nivel
- Lenguaje de Bajo Nivel
- Lenguaje de Alto Nivel

11. Según la Clasificación de Registros en Lenguaje Ensamblador, elija los registros visibles al Usuario

(Manipulados por el usuario):

- Registro de Propósito General
- Registros Apuntadores e índices
- Registros de Segmento
- Registros de Bandera

12. Empareje cada sección de memoria con el Segmento que corresponde.

Segmento Extra: Sección de memoria utilizada ocasionalmente para algunas instrucciones de cadenas o para ampliar la capacidad de los registros

Segmentos de Datos: Sección de memoria que contiene la mayor parte de datos utilizados por el programa

Segmento de código: Sección de memoria que tiene las instrucciones y toda la lógica utilizada para el programa

Segmento de Pila: Sección de memoria destinada para el arreglo de pila con que trabaja el microprocesador

13. Las computadoras únicamente pueden interpretar el paso o interrupción de corriente eléctrica, por lo cual el sistema hexadecimal se ajusta a las necesidades de la programación

R/Falso

14. Una dirección de memoria y una dirección de desplazamiento generan una dirección de segmento

R/Falso (Genera una dirección de memoria)

15. Seleccione los registros Banderas disponibles en el Lenguaje Ensamblador

- Acarreo
- Desbordamiento
- Paridad
- Cero
- Interrupción
- Signo

16. Según la clasificación por su Ejecución, ¿Que lenguajes de programación existen?

- Lenguaje Compilado
- Lenguaje Interpretado

17. Elija los componentes de la Unidad Central de Proceso.

- Registros
- Unidad Arimético-Lógica
- Unidad de Control

18. Un nemónico es una cadena de '0' y '1'

R/Falso

19. Seleccione los grupos de palabras reservadas que se utilizan en el Lenguaje Ensamblador

- Operadores
- Simbolos Predefinidos
- Directivas
- Nemónicos

20. Empareje cada subclasificación de lenguajes de Programación con su ejemplo específico

Lenguaje de Bajo Nivel: Lenguaje Ensamblador

Lenguaje escrito en código binario: Lenguaje de Maquina

Lenguaje de Medio Nivel: C++

Lenguaje de Alto Nivel: C#

21. Analice el siguiente programa en Lenguaje Ensamblador y elija la opción del resultado que se mostrara en pantalla.

```
.model small  
.code  
programa:  
    mov AX, @DATA  
    MOV DS, AX  
    MOV AX, 00H  
    MOV BX, 00H  
    MOV AL, 42H ;"B"  
    MOV DL, AL  
    MOV AH, 02H  
    INT 21H  
    MOV AH, 4CH  
    INT 21H  
    .STACH  
END
```

R/ Se mostrará una B en pantalla

22. Empareje cada concepto con el elemento básico correspondiente a un Programa en Lenguaje Ensamblador

Instrucciones: Conjunto de nemónicos e identificadores validos que ejecutan una acción

Modelos: Indican los tamaños del programa.

Identificadores: Nombres que se le dan a los elementos de un programa, indicados por el programador

Directivas: Inician los segmentos, indican al Ensamblador la estructura del programa

23. Empareje cada definición con el Modelo del Programa en Lenguaje Ensamblador correspondiente

Compact: Datos>= un segmento (si se llena un segmento se crea otro), código un solo segmento (64k bytes por cada segmento)

Tiny: Código y datos en el mismo segmento (64k bytes total)

Small: Código y datos en segmentos diferentes (64k bytes c/u)

Medium: Código >= un segmento (si se llena un segmento se crea otro=, datos un solo segmento (64k bytes por cada segmento)

Large: Código y datos >= un segmento (Si se llena un segmento se crea otro)

24. Analice el siguiente programa en Lenguaje Ensamblador y elija la opción del resultado que se mostrara en pantalla.

```
.model small  
  
.code  
  
programa:  
  
    mov AX, @DATA  
  
    MOV DS, AX  
  
    MOV AX, 00H  
  
    MOV BX, 00H  
  
    MOV AL, 42  
  
    MOV DL, AL  
  
    MOV AH, 02H  
  
    INT 21H  
  
    MOV AH, 4CH  
  
    INT 21H  
  
.STACK  
  
END
```

R/ Se mostrará un * en pantalla

25. El Instruction Register contiene la siguiente instrucción a ejecutar

R/Falso

26. Según la clasificación por su Paradigma. ¿Cuáles categorías de lenguajes podemos encontrar?

-Lenguaje Imperativo

-Lenguaje Declarativo

27. Elija los componentes de la Estructura Básica de Modelo Von Neuman

-Sistemas de Interconexión (Buses)

- Entrada/Salida
- Unidad Central de Procesos
- periféricos
- Memoria

28. Nintendo Entertainment System se programó en Lenguaje Ensamblador en su totalidad

R/Verdadero

29. El Instruction Pointer no es visible al usuario, pero sí modificable por el mismo, siendo un registro de control y estado.

R/ Verdadero

30. Analice el siguiente programa en Lenguaje Ensamblador y elija la opción del resultado que se mostrara en pantalla.

.model small

.code

programa:

mov AX, @DATA

MOV DS, AX

MOV AX, 00H

MOV BX, 00H

MOV AL, 50

MOV DL, AL

MOV AH, 02

INT 21H

MOV AH, 4CH

INT 21H

MOV AH, 00H

MOV AL, 03H

INT 10H

.STACK

END

R/ Se mostrará un 2 en pantalla

31. Analice el siguiente programa en Lenguaje Ensamblador y elija la opción del resultado que se mostrará en pantalla.

```
.model small  
  
.code  
  
programa:  
  
    mov AX, @DATA  
  
    MOV DS, AX  
  
    MOV AX, 00H  
  
    MOV BX, 00H  
  
    MOV AL, 55H ; "U"  
  
    MOV DL, AL  
  
    MOV AH, 02  
  
    INT 21H  
  
    MOV AH, 00H  
  
    MOV AL, 03H  
  
    INT 10H  
  
    MOV AH, 4CH  
  
    INT 21H  
  
.STACH  
  
END
```

R/ Se mostrará una U en la pantalla.

32. Los lenguajes que se encuentran más cercanos a los programadores y usuarios se denominan lenguajes de bajo nivel, debido a su bajo nivel de abstracción (alto nivel)

R/ Falso

33. La simplicidad de programación es una característica del Lenguaje Ensamblador.

R/ Falso

34. Elija los dispositivos Periféricos

-Monitor -Teclado -Mouse -Impresora -Cámara Web

35. Lo programado en Lenguaje Ensamblador se traduce en Lenguaje de Máquina por medio de un Ensamblador

R/Verdadero

36. El lenguaje de Máquina evolucionó en el Lenguaje Ensamblador.

R/ Verdadero

37. Los Registros trabajan directamente con la CU y se mueven a la velocidad del ALU, teniendo capacidades de almacenamiento permanentes

R/Falso

38. Según la clasificación de Registros en Lenguaje Ensamblador, elija los Registros de Control y Estado

(manipulados por el microprocesador)

- Registro de Dirección de Memoria (MAR)
- Registro de Apuntador de instrucciones (IP)
- Registro de Instrucción (IR)
- Registro de Dato de Memoria (MDR)

PARCIAL 2

1. En lenguaje Ensamblador para el modelo de procesador intel 8086, el error "jump destination too far

by XX bytes" puede solventarse con:

- Reestructuración del código fuente
- Utilizando saltos intermedios

2. Tanto los macros como los procedimientos deben definirse previos a ser referenciados

-Falso

3. En la memoria en sus primeros 1024 bytes (000h - 3FFh), ¿Qué se almacena?

-Vector de interrupciones

4. A un arreglo multidimensional se le conoce como Matriz

-verdadero

5. La Arquitectura Clásica del Computador enfoca sus características en el procesamiento, almacenamiento

y transferencia de datos

-verdadero

6. La diferencia entre un Procedimiento y un Macro es que el primero copia la porción de código y se continua

la ejecución secuencial, mientras que el segundo modifica el IP para ejecutar la posición de código específico

y luego retomar el punto en que se interrumpió la secuencia.

-Falso

7. La portabilidad es una ventaja del lenguaje ensamblador

-Falso

8. Empareje cada palabra reservada, en Lenguaje Ensamblador, con su función respectiva:

-Utilizada para hacer llamada a procedimientos definidos CALL

-Utilizada para declarar el fin de la definición de un programa END

-Utilizada para indicar, al ensamblador, el momento de salto al punto donde fue llamado del procedimiento y continuar con la ejecución secuencial RET

-Utilizada para indicar, al ensamblador, el momento de salto al punto de ejecución de interrupción y continuar con la ejecución secuencial IRET

-Utilizada para la definición de procedimientos PROC

-Utilizada para declarar el fin de la definición de un procedimiento. ENDP

9. A la agrupación de instrucciones que se repiten constantemente a lo largo de un programa, en Lenguaje

de Maquina, se le denomina "Macros"

-Falso

10. Los Registros trabajan directamente con la CU y se mueven a la velocidad del ALU, teniendo capaci-

dades de almacenamiento permanente

-Falso

11. Cada posición de una cadena de caracteres, equivale a una posición en un vector de bajo nivel y no un espacio reservado de memoria

-Falso

12. La arquitectura Harvard se le conoce como la arquitectura Clásica de Computador.

-Falso

13. En Lenguaje Ensamblador, podemos definir parámetros en los Procedimientos, pero para los Macros no,

con los últimos se debe modificar previamente el entorno.

-Falso

14. El instructor Pointer no es visible al usuario, pero si modificable por el mismo, siendo un registro

de control y estado

-Verdadero

15. Cada segmento definido es de 64 bytes de memoria

-Falso

16. La diferencia entre un Macro y un procedimiento es que el primero copia la porción de código y se

continúa la ejecución secuencial, mientras que el segundo modifica el IP para ejecutar la porción de código específico y luego retornar al punto en que se interrumpió la secuencia

-Verdadero

17. Ordene secuencialmente los eventos que se dan al ejecutar una interrupción en Lenguaje Ensamblador

- | | |
|---|----------------|
| -Modificar la dirección del ISR, que corresponde a la interrupción, al IP | Tercer evento |
| -Ejecutar instrucciones del ISR hasta encontrar la palabra reservada "IRET" | Cuarto evento |
| -Finalizar la instrucción previa al detectar la interrupción | Primer evento |
| -Sacar de Pila el CS, IP y Banderas | Quinto evento |
| -Insertar en Pila el CS, IP y Banderas | Segundo evento |

18. El ensamblador traduce lo programado en Lenguaje de Máquina a Lenguaje Ensamblador

-Falso

19. Un nemónico es una cadena de "0", "1"

-Falso

20. El Instruction Register contiene la siguiente instrucción a ejecutar

-Falso

21. Los lenguajes más cercanos a la arquitectura de hardware se denominan lenguajes de alto nivel,

debido a su nivel de complejidad

-Falso

22. Las computadoras únicamente pueden interpretar el paso o interrupción de corriente eléctrica por lo cual, el sistema hexadecimal se ajusta a las necesidades de la programación

-Falso

23. Tomando la formula de Mapeo Lexicográfico.

Dirección efectiva = Dirección de Memoria + (fila i * tamaño de un elemento) + (columna j * número de elementos por columna * tamaño de elemento)

El almacenamiento en memoria, de una Matriz bidimensional M[i,j], se dio por:

-Columnas

24. La simplicidad de programación es una característica del Lenguaje Ensamblador

-Falso

25. El lenguaje de Maquina evoluciona en el Lenguaje Ensamblador

-Verdadero

26. Empareje cada concepto con el elemento básico correspondiente a un Programa en Lenguaje Ensamblador

-Dan nicio a los segmentos, indican al Ensamblador la estructura del programa	Directivas
-Indican los tamaños del programa, es decir, la cantidad máxima de código y datos	Modelos
-Conjuntos de nemónicos e identificadores válidos que ejecutan una acción	Instrucciones
-Nombres que se le dan a los elementos de un programa, indicados por el programados	Identificadores

27. Empareje cada definición con el tipo de interrupcion correspondiente

-Interrupción provocada por el ensamblador, emplea las Interrupción de DOS de software funciones del Sistema Operativo para la manipulación de Hardware

-Interrupción generada por dispositivos periféricos Interrupción Externa de Hardware

-Interrupción generada por eventos ocurridos durante la ejecución del programa, manejados completamente por el hardware Interrupción interna de Hardware

-Interrupción provocada por el ensamblador, contiene las rutinas de I/O y tablas que indican los estados de los dispositivos del sistema Interrupción de BIOS de Software

28. El caché y la Memoria Principa(RAM) son los elementos mas veloces de la computadora

-Falso

29. A la colección de instrucciones que le dan tratamiento a una interrupción, se le conoce como

-ISR

30. Un método lleva un salto en forma implicita, indicando el retorno al punto de salto inicial

por medio de la palabra reservada RET

-Verdadero

31. Tomando la formula de Mapeo Lexicográfico.

$$\text{Dirección efectiva} = \text{Dirección de Memoria} + (\text{fila } i * \text{tamaño de elementos por fila} * \text{tamaño de elementos}) + (\text{columna } j * \text{tamaño de elemento})$$

El almacenamiento en memoria, de una Matriz bidimensional $M[i,j]$, se dio por:

-Filas

32. A la colección ordenada de datos de cualquier tipo, que poseen el mismo formato en sus datos

consecutivos se le conoce como Arreglo

-Falso

33. Lo programado en Lenguaje Ensamblador se traduce en Lenguaje de Máquina por medio de un Ensamblador

-Verdadero

34. Selecciones algunas de las ventajas que proporcionan los Métodos al utilizarse en la programación

en Lenguaje Ensamblador

- Reducen la cantidad de código
- Permiten la reutilización de código
- Permiten la organización y modularización del programa
- Simplifican el mantenimiento de código

35. A la obtención de elementos en un arreglo multidimensional por medio de una fórmula matemática

definida se le conoce como Mapeo Lexicográfico

-Verdadero

36. El ensamblador traduce lo programado en Lenguaje de Máquina a Lenguaje Ensamblador

-Falso

37. Al referenciar un Macro, en tiempo de ensamblaje, se sustituye la llamada por conjunto de instrucciones que contiene

-Verdadero

38. Elija los componentes de la unidad central de proceso

- Registros
- Unidad Aritmético-Lógica
- Unidad de control

39. Empareje cada especificación con el Registro correspondiente

- | | |
|--|------------------------------|
| -Usualmente conserva la base de los datos que hay en la memoria | Registro Base(BX, EBX) |
| -Usualmente conserva el resultado temporal después de una operación aritmética o lógica | Registro Acumulador(AX, EAX) |
| -Usualmente contiene el conteo de ciertas instrucciones para corrimientos y rotaciones del número de bytes o contador LOOP | Registro Contador(CX, ECX) |
| -Usualmente contiene la parte mas significativa del producto luego de una multiplicación o del dividendo antes de una división; de uso general | Registro de Datos(DX, EDX) |

40. Cada posición de una cadena de caracteres, equivale a una posición en un vector de bajo nivel y no un espacio reservado en memoria

- Falso