



EXAMEN FINAL

Actividad: Codificación en ensamblador ix86

Objetivo de aprendizaje.

Realiza programación en lenguaje ensamblador de la arquitectura R2 ix86, ix386 o ix64 para realizar operaciones de procesamiento matemático o textual sencillo aplicando diferentes modos de direccionamiento.

Pasos para realizar la actividad.

1. Cada alumno deberá seleccionar un (1) problema de la lista que se provee en la sección [Problemas] al final de la guía. El criterio a utilizar es que una vez que el alumno haya seleccionado su problema a resolver, este ya no podrá ser elegido por otro alumno (selección única).
2. Selecciona el entorno de desarrollo a emplear para codificar tu programa en ensamblador. La opción propuesta es: GUI TurboAssembler. No aplica otro software.
3. Planifica (algoritmiza) la solución a tu problema, codifica, prueba y valida tu programa.
4. Una vez esté correcto, entrega tu producto:
 - a. Informe del proceso con resultado y conclusión (pantallas de captura del código y resultado de la salida del programa).
 - b. Código fuente (incluye todos los archivos fuente empleados, incluyendo algoritmo, si se realizó).
 - c. Ejecutable.

[PROBLEMAS]

- 1) **Multiplicación de dos enteros por sumas sucesivas.** Los factores se reciben en dos registros o variables de memoria y el resultado debe quedar en un tercer registro o variable e imprimir en pantalla. Valide en caso de que cualquiera de los factores sea cero. Optimice el ciclo usando el factor más pequeño como control de iteraciones.
- 2) **División de dos enteros por restas sucesivas.** El dividendo se recibe en un registro y el divisor en otro (o en variables de memoria). El cociente debe quedar en un tercer registro y el residuo en otro (o en variables de memoria) e imprimir en pantalla. Valide en caso de que el dividendo sea cero. En caso de que el divisor sea cero, se debería generar una excepción, lo cual no es requerido en este momento implementar.

- 3) **Determinación del Máximo Común Divisor (MCD) y el Mínimo Común Múltiplo (mcm).** Dados dos números se reciben en dos registros, determinar el MCD empleando el algoritmo de Euclides y almacenar su resultado en un tercer registro. Finalmente, determinar el mcm multiplicando los números y dividiéndolos entre el MCD. Almacenar el resultado en un cuarto registro e imprimir en pantalla.
- 4) **Determinar las sumatoria de números naturales consecutivos.** El programa debe recibir dos números y calcular las sumatoria de todos los números consecutivos entre estos. Por ejemplo, si se proveen 5 y 12, el programa debe ser capaz de calcular la suma de los números $5 + 6 + 7 + 8 + 9 + 10 + 11 + 12$ y entregar su suma = 68, e imprimir en pantalla. Como ayuda se sugiere investigar las series notables.
- 5) **Determinar las sumatoria de números naturales impares consecutivos.** El programa debe recibir dos números impares y calcular las sumatoria de todos los números impares consecutivos entre estos. Por ejemplo, si se proveen 5 y 19, el programa debe ser capaz de calcular la suma de los números $5 + 7 + 9 + 11 + 13 + 15 + 17 + 19$ y entregar su suma = 96, e imprimir en pantalla. Como ayuda se sugiere investigar las series notables.
- 6) **Determinar las sumatoria de los cuadrados de los primeros n números naturales consecutivos.** El programa debe recibir el último número de la serie de cuadrados a los cuales se va a calcular su suma. Por ejemplo, si se provee 5 el programa debe ser capaz de calcular la suma de los cuadrados de los números naturales del 1 al 5: $1 + 2^2 + 3^2 + 4^2 + 5^2$ y entregar el resultado = 55, e imprimir en pantalla. Como ayuda se sugiere investigar las series notables.
- 7) **Determinar si un arreglo de caracteres es palíndromo.** El arreglo de caracteres se almacena en una variable de memoria con nombre "cadena". El tamaño del arreglo se recibe otra variable. El resultado debe quedar en un registro: 1 si es palíndromo y 0 si no lo es, e imprimir en pantalla la cadena y el resultado. Se sugiere usar dos punteros: uno al inicio del arreglo y otro al final, de forma tal que se vayan acercando en cada ciclo de procesamiento, mientras el puntero del final sea mayor que el de inicio.
- 8) **Invertir una cadena.** Considere que la cadena corresponde a un arreglo de caracteres almacenado como variable de memoria con nombre "cadena". El tamaño de la cadena se recibe en otra variable o se determina. Se sugiere usar dos punteros, uno al inicio de la cadena y otro al final e irlos acercando. Debe imprimir en pantalla la cadena original y la invertida al final.

- 9) **Conteo de ocurrencias de un carácter.** El programa debe recibir una cadena y un carácter a buscar. El tamaño de la cadena se recibe en una variable o se determina. El programa debe recorrer la cadena verificando coincidencias de sus caracteres con el carácter buscado y contando las ocurrencias del mismo. Al final debe retornar en pantalla cuántas ocurrencias se dieron.
- 10) **Determinar si una cadena es subcadena de otra.** El programa de recibir dos cadenas la primera es la fuente y la segunda el patrón buscado, y debe retornar la posición de la cadena fuente donde inicia el patrón o -1 si no se encuentra. El programa debe recorrer la cadena fuente buscando la primera ocurrencia del primer carácter de la cadena patrón. Si la encuentra, verifica si a partir de ese carácter se encuentra el patrón. De ser así retorna la posición. Si no coincide, continúa la búsqueda del primer carácter a partir de la siguiente posición, hasta que se llegue al final de la cadena fuente. Debe imprimir en pantalla la cadena original, la subcadena y el resultado.
- 11) **Reemplazo de caracteres.** El programa debe recibir una cadena, un carácter a buscar y un carácter reemplazo. Debe recorrer la cadena y reemplazar cada carácter de ésta que coincida con el carácter buscado con el carácter reemplazo. Debe imprimir en pantalla la cadena original, los caracteres a buscar y reemplazo y la cadena final.
- 12) **Reemplazo de caracteres con Complemento a 9.** El programa debe recibir una cadena y debe recorrerla buscando caracteres correspondientes a los dígitos decimales (del 0 al 9) y debe reemplazar cada dígito encontrado por su complemento a 9. Debe imprimir en pantalla la cadena original y la cadena final.
- 13) **Dados dos arreglos A y B de números enteros, los sume y guarde el resultado en el arreglo B.** El tamaño de los arreglos se recibe en una variable.
- 14) **Dado un arreglo de números enteros, sume los números pares.** El tamaño del arreglo se recibe en una variable. Debe imprimir en pantalla el resultado.
- 15) **Dado un arreglo de números enteros, sume los números impares.** El tamaño del arreglo se recibe en una variable. Debe imprimir en pantalla el resultado.
- 16) **Dado un arreglo de números enteros, sume los números en las posiciones pares.** El tamaño del arreglo se recibe en una variable. Debe imprimir en pantalla el resultado.
- 17) **Dado un arreglo de números enteros, determine cuál es el menor de sus elementos.** El tamaño del arreglo se recibe en una variable. Debe imprimir en pantalla el resultado.

- 18) **Dado un arreglo de números enteros, determine cuál es el mayor de sus elementos.** El tamaño del arreglo se recibe en una variable. Debe imprimir en pantalla el resultado.
- 19) **Dado un arreglo de números enteros, lo ordene de menor a mayor.** El tamaño del arreglo se recibe en una variable.
- 20) **Dado un arreglo de números enteros, calcule la media aritmética de sus elementos.** El tamaño del arreglo se recibe en una variable. Debe imprimir en pantalla el resultado.
- 21) **Dado un arreglo de números enteros correspondientes a calificaciones, determine el número de aprobados (calificación ≥ 60) y de reprobados.** El tamaño del arreglo se recibe en una variable. Debe imprimir en pantalla el resultado.
- 22) **Dado un arreglo de números enteros A determine si sus elementos corresponden a años bisiestos y escriba en el arreglo de enteros B en la posición correspondiente de cada elemento de A, un 1 si el año es bisiesto y un 0 si no.** El tamaño del arreglo se recibe en una variable. Un año es bisiesto si: a) es divisible exacto de 100 y es múltiplo de 400 o b) no es divisible entre 100 y es múltiplo de 4.
- 23) **Cree un procedimiento simple de encriptado y desencriptado** que aplique estas operaciones sobre una cadena de caracteres almacenada en un arreglo y cuyo tamaño está en una variable. La clave de encriptamiento está en otra variable. El procedimiento debe usar la función XOR que es perfectamente reversible para encriptar-desencriptar (Si dato XOR clave = encriptado \square encriptado XOR clave = dato). Nota: La clave de encriptamiento debe ser mayor que 0 y menor o igual a 255.
- 24) **Genere los primeros “n” valores de la serie Fibonacci** en un arreglo de enteros almacenado en la variable de memoria “Fibonacci”. Considere que el valor de n está almacenado en un registro o variable.
- 25) **Genere los primeros “n” valores de la serie Triangular** en un arreglo de enteros almacenado en la variable de memoria “triangular”. Considere que el valor de n está almacenado en un registro o variable. La serie triangular es la serie donde $a_k = \sum_{i=1}^k n_i$, es decir, cada elemento es la suma de los números naturales anteriores y él inclusive. Por ejemplo, el primer elemento de la serie es 1, el segundo es 3 (1 + 2), el tercero es 6 (1 + 2 + 3), y así sucesivamente.
- 26) **Genere los primeros “n” valores de la serie de cuadrados** en un arreglo de enteros almacenado en la variable de memoria “cuadrados”. Considere que el valor de n está almacenado en un registro o variable. La serie cuadrados se forma por el cuadrado del ordinal del elemento: 1, 4, 9, 16, ...

- 27) **Genere los primeros “n” valores de la serie alternante de signo e incremento - decremento** en un arreglo de enteros almacenado en la variable de memoria “alterna”. Considere que el valor de n está almacenado en un registro o variable. La serie alternante de signo e incremento-decremento, inicia en +1 y en el próximo elemento alterna el signo e incrementa cada número par en 1 y decrementa cada número impar en 1: +1, -3, +2, -5, +4,
- 28) **Dado un arreglo de números enteros, lo ordene de mayor a menor.** El tamaño del arreglo se recibe en una variable.
- 29) **Determinar si un número es primo.** El programa recibe un número natural y debe verificar se es primo o no, indicando esto en su respuesta en pantalla.
- 30) **Calcula los cuadrados de un vector.** El programa recibe el vector en un arreglo de números enteros y calcula el cuadrado de cada elemento y lo almacena en un segundo arreglo en la posición del índice del elemento correspondiente.
- 31) **Alternar mayúsculas con minúsculas.** El programa recibe una cadena de caracteres y lo recorre intercambiando los caracteres alfabéticos en mayúsculas con su contraparte en minúscula y viceversa. Debe imprimir en pantalla el resultado.
- 32) **Convertir a mayúsculas.** El programa recibe una cadena de caracteres y lo recorre dejando todos los caracteres alfabéticos en mayúsculas. Debe imprimir en pantalla el resultado.
- 33) **Convertir a minúsculas.** El programa recibe una cadena de caracteres y lo recorre dejando todos los caracteres alfabéticos en minúsculas. Debe imprimir en pantalla el resultado.
- 34) **Codificar jerigonza.** El programa recibe una cadena de caracteres y lo recorre codificando sus palabras según las reglas de la jerigonza: agregando a cada sílaba una nueva enseguida conformada por la letra p seguida de la vocal de la sílaba. Por ejemplo, si la cadena es “Hola Mundo”, su correspondiente en jerigonza sería: “Hopolapa Munpudopo”. Debe imprimir en pantalla el resultado.
- 35) **Codificar en Malespín.** El programa recibe una cadena de caracteres y lo recorre codificando sus palabras a Malespín (intercambiando: a por e, i por o, b por t, f por g y p por m). Por ejemplo, si la cadena es “Hola Mundo” el resultado en malespín sería “Hile Pundi”. Debe imprimir en pantalla el resultado.

Criterios de evaluación.

- Entregar todo lo solicitado en el punto #4 de los pasos para realizar la actividad.
- Se valorará tanto el proceso seguido como el resultado. El programa ejecutable debe funcionar y realizar lo que se requiere. En caso de dificultad de solicitar entrada, se aceptará emplear valores fijos en el programa. El programa debe mostrar el resultado por pantalla.
- Pueden auxiliarse de Internet y deberán incluir en el documento las referencias bibliográficas (las páginas que utilizaron para dar solución a sus problemas). Incluirlo en el documento.
- La selección del problema es única, no se admite la selección de problemas repetidos.
- La convención para el nombre que le pondrán al programa es: la palabra "PROGRAMA", seguido del número de programa. Por ejemplo: **PROGRAMA5**
- Solo se aceptan documentos en WORD o PDF únicamente.
- Esperar la confirmación de parte de docente, si lo envían ya sea por correo electrónico o bien por WhatsApp.
- Fecha tope de envío: el día que el departamento de ASA calendarice el examen final.
- Para el control de la elección del problema por parte de los alumnos, deberán de indicármelo por WhatsApp (85359915), con los siguientes datos:

Asignatura: AMC II

Nombre del alumno: XXXXX YYYYY

Programa elegido: ###

En este caso yo les confirmare si el problema está disponible o ya ha sido seleccionado.