

Guía de Ejercicios Prácticos

Assembler INTEL 8086

75.03-95.57 Organización del Computador

- 1.** Escribir un programa que imprima por pantalla “Organización del Computador”.
- 2.** Realizar un programa en assembler Intel x86 que imprima por pantalla la siguiente frase: “El alumno [Nombre] [Apellido] de Padrón N° [Padrón] tiene [Edad] años para esto se debe solicitar previamente el ingreso por teclado de:
 - Nombre y Apellido
 - N° de Padrón
 - Fecha de nacimiento
- 3.** Realizar un programa que resuelva X^Y teniendo en cuenta que tanto X e Y pueden ser positivos o negativos.
- 4.** Escribir un programa que lea 15 números ingresados por teclado. Se pide imprimir dichos números en forma decreciente.
- 5.** Dado un vector de 20 números almacenados en el formato BPF c/s de 16 bits, escriba un programa que calcule el máximo, mínimo y el promedio de los números e imprimirlos por pantalla.
- 6.** Dado un vector de 30 números decimales (positivos/negativos) se pide escribir un programa que invierta el vector, es decir, el último elemento queda en el primer lugar, el anteúltimo en el segundo, etc.
- 7.** Dada una matriz de 3x3 de números almacenados en BPF c/s de 16 bits, calcule la traza e imprimirla por pantalla.
- 8.** Dada una matriz de 4x4 de números almacenados en BPF c/s de 16 bits, calcule la diagonal inversa e imprimirla por pantalla.
- 9.** Dada una matriz de 5x5, determinar si dicha matriz es triangular superior y/o inferior e imprimir el resultado por pantalla.
- 10.** Realizar una rutina interna que reciba como parámetro un campo PACK en formato de Decimal Empaquetado de 2 bytes y devuelva en un campo RESULT en formato carácter de 1 byte, indicando una ‘S’ en caso que sea un empaquetado válido, y en caso contrario una ‘N’.
- 11.** Realizar una rutina interna que reciba como parámetro un campo DIA en formato de caracteres de 2 bytes y devuelva en un campo resultado RESULT en formato carácter de 1 byte, indicando una ‘S’ en caso que el día sea válido, y en caso contrario una ‘N’.
Los valores válidos son LU, MA, MI, JU, VI, SA, DO.
- 12.** Realizar una rutina interna que reciba como parámetro un campo MES en formato BPF c/s de 8 bits y devuelva en un campo resultado RESULT en formato carácter de

1 byte, indicando una 'S' en caso que el valor del mes sea válido, y en caso contrario una 'N'.

- 13.** Se tiene una fecha en formato carácter DD/MM/AAAA se pide realizar una rutina interna que realice la validación dejando en el campo RESULT de 1 byte una 'S' si es válida o una 'N' en caso contrario.
- 14.** Realizar una rutina interna que reciba como parámetros las direcciones (DIR1 y DIR2) de dos campos hexadecimales de 2 bytes de longitud cada uno y realice la suma de ambos (en BPF s/signo de 16 bits) dejando el resultado en el campo resultado RESULT en formato BPF c/s 16 bits.
- 15.** Se lee de un archivo una serie de números en formato carácter de 3 bytes. Se pide realizar un programa que realice la sumatoria de esos números e informe el resultado por pantalla, indicando además la cantidad de números válidos e inválidos leídos del archivo.
- 16.** Se tiene el campo NUM que corresponde a un número almacenado como BPF c/s 16 bits en memoria. Se pide realizar un programa necesario para rebatir los bits que componen al número e imprimir el número en configuración binaria por pantalla. (Rebatir significa poner el 1er. bit del número en el lugar del último, el 2do. en el anteúltimo y así sucesivamente)
- 17.** La liga de basquetball de Villa Tachito desea tener un programa para determinar quién fue el ganador del torneo anual de clubes. Para ello cuenta con un archivo que contiene:
- Nombre del equipo: 20 caracteres alfanuméricos
 - Resultados: BPF S/S de 16 bits
 - Tantos a favor: Empaquetado de 2 bytes
 - Tantos en contra: Empaquetado de 2 bytes

El programa deberá imprimir por pantalla el nombre del campeón del torneo junto con la cantidad de partidos ganados y perdidos y su diferencia de tantos. Para determinar la cantidad de partidos ganados se procesa el campo Resultados que indica por cada bit el resultado de un partido del torneo. (En total cada equipo jugó 16 partidos) Si el bit está en 1 significa que el equipo ganó ese partido, si está en 0 significa que lo perdió.

- 18.** Se tienen las siguientes definiciones de rótulos en un programa assembler Intel x86:
- | | |
|----------|--------|
| STRING_A | RESD 1 |
| STRING_B | RESD 1 |
| CHAR | RESB 1 |

Las variables STRING_A y STRING_B tienen la dirección de inicio de un string (cadena de caracteres terminados con un byte x'00') mientras que la variable CHAR contiene un carácter en codificación ASCII. Se pide:

- a. Realizar una rutina interna que reciba como parámetros a los rótulos STRING_A y STRING_B y devuelva un byte RESULT1 con una 'S' si los dos strings son iguales o una 'N' en caso contrario.
- b. Realizar una rutina interna que reciba como parámetros al rótulo STRING_A y CHAR y devuelva una FULLWORD con la cantidad de ocurrencias del carácter CHAR dentro del string, en BPF c/s de 32 bits.

19. Se cuenta con una matriz (M) de 20x20 cuyos elementos son BPFC/S de 16 bits y un archivo (carbina.dat) cuyos registros están conformados por los siguientes campos:

- Cadena de 16 bytes de caracteres ASCII que representa un BPFC/s de 16 bits
- BPF s/s de 8 bits que indica el número de fila de M
- BPF s/s de 8 bits que indica el número de columna de M

Se pide codificar un programa que lea los registros del archivo y complete la matriz con dicha información. Como el contenido de los registros puede ser inválido deberá hacer uso de una rutina interna (VALREG) para validarlos (los registros inválidos serán descartados y se procederá a leer el siguiente). Luego realizar la sumatoria de la diagonal secundaria e imprimir el resultado por pantalla.

Nota: Se deberá inicializar M con ceros por si no se lograra completar todos los elementos con la información provista en el archivo.

20. Se dispone de una matriz C que representa un calendario de actividades de una persona. La matriz C está formada por 7 columnas (que corresponden a los días de la semana) y por 6 filas (que corresponden a las semanas que puede tener como máximo un mes en un calendario). Cada elemento de la matriz tiene una longitud de 200 caracteres que a las actividades a realizar ese día (como mucho se podrán poner 10 actividades).

Además se dispone de un archivo de entrada llamado CALEN.DAT para cargar el contenido de los vectores de la matriz C con la siguiente información:

- Día de la semana: Carácter de 2 bytes (DO, LU, MA, MI, JU, VI, SA)
- Semana: Binario de 1 byte (1..6)
- Actividad: Caracteres de longitud 20 con la descripción.

Como la información leída del archivo puede ser errónea, se dispone de una rutina interna llamada VALCAL para su validación. La rutina recibirá como parámetros el registro de entrada y un 1 byte en donde devolverá una 'S' si el registro es válido, o en caso contrario una 'N'. Se deberá validar el Día de la semana y la Semana.

Se pide realizar un programa assembler Intel x86 que cargue la matriz C con la

información del archivo de entrada previamente validada. Luego se solicita listar todas las actividades de todas las semanas del día almacenado en Dialng DD XX.

21. Se cuenta con un archivo en formato binario llamado ENCUESTA.DAT que contiene información de las respuestas de una encuesta. La encuesta consultaba a qué candidato votaría (de una lista de 4 opciones) y se realizó en 10 ciudades.

Cada registro del archivo representa una respuesta y contiene la siguiente información:

- Código de candidato: 2 bytes en formato ASCII (AF, MM, RL, SM)
- Código de ciudad: 1 bytes en formato binario punto fijo sin signo

Se pide realizar un programa en assembler Intel x86 que lea el archivo y por cada registro llene una matriz (M) de 4x10 donde cada fila representa a un candidato y cada columna una ciudad. Cada elemento de la matriz M representa la sumatoria de respuestas para cada candidato en cada ciudad; para el llenado de M se hará uso de una rutina interna VALREG q validará los datos de cada registro descartando los incorrectos.

Por último el programa debe pedir ingresar un código de candidato e informar por pantalla en que ciudad tiene menos intención de votos teniendo en cuenta que habrá un vector en memoria de longitud 10, cuyos elementos tienen 20 bytes de longitud con los nombres de las ciudades.