

Nombre: Monica Elizabeth Alba Gonzalez Clave única: 160502 Calif: 9.40

Cada una de las preguntas: La 1 vale 3.0 pts., el 3 vale 2.0 pts y los restantes 2 y 4, valen 2.5 pts cada uno.  
**CONTESTAR CON TINTA.** Duración 1 hs 55 min o de 7-8:55 hs.

1.- Elaboración de programa en ensamblador

En el archivo "SgExm18.asm" se encuentran los siguientes dos arreglos de valores enteros, *arrInd* y *arrPot*. Así, *arrPot* es un arreglo de potencias eléctricas, expresadas en KVA y *arrInd* es un conjunto de índices que indican la posición individual de las potencias eléctricas: p.e. *arrInd*[0] cuyo valor es 7, indica a la potencia eléctrica *arrPot*[7], que contiene el valor 11000; *arrInd*[2] de valor 8, indica a *arrPot*[8], que contiene el valor 10000.

Lo mostrado en el segmento .DATA, de la figura 1, es un ejemplo particular, pero usted deberá programar tomando en cuenta que *arrInd* y *arrPot* puedan tener cualquier cantidad de valores y de líneas DWORD. Estas cuatro etiquetas deberán mantenerse continuas como aparece en la figura 1.

```
.DATA
arrInd DWORD 7, 2, 8, 11
        DWORD 14, 3, 10, 6, 0
        DWORD 16, 9, 1, 4, 5
totInd  SDWORD ? ; Total de Indices
arrPot  DWORD 1000, 2000, 3000, 4000
        DWORD 5000, 0, 12000, 11000
        DWORD 10000, 9000, 8000, 7000
totPot  SDWORD ? ; Total de Potencias
```

Figura 1

Recuerde que los arreglos comienzan con el índice 0 (cero). Las etiquetas *totInd* y *totPot*, representan respectivamente, el total de índices en el arreglo *arrInd*, y el total de potencias eléctricas en *arrPot*.

Usted deberá programar cálculos, procedimientos e impresiones, para desplegar los resultados de la figura 2:

```
Soy 967456AdrianB

Total de indices: 14
Total de potencias: 12

Posición 7 con valor: 11000
Posición 2 con valor: 3000

Posición 5 con valor: 0

HASTA LUEGO.
```

Figura 2

CONTINÚA EN LA SIGUIENTE PÁGINA. >>>

Para lo cual deberá, desde *main*:

- a) Invocar a un procedimiento *ID* que imprima su “identificación” de alumno formada por su clave única seguido de su nombre y la inicial de su apellido, p.e. “967456AdrianR”;
- b) Llamar al procedimiento *CtaElemArrD* que permita contar los elementos de un arreglo *DWORD*, y regrese como resultado dicha cuenta. Primero lo hará para calcular el total de índices, que imprimirá el mismo *main*.
- c) Por segunda vez, invocar al procedimiento *CtaElemArrD*. Ahora lo hará para calcular el total de potencias eléctricas, que imprimirá desde *main*.
- d) Llamar al procedimiento *ImprimirArrD* para que imprima el contenido del elementos de *arrPot* en el orden indicado por el arreglo *arrInd*.
- e) Al final imprimir HASTA LUEGO.

El archivo “.asm” deberá renombrarlo con su “identificación”, p.e. “967456AdrianR18.asm” poniéndolo dentro de un archivo zip “967456AdrianR18.zip” y al terminarlo deberá subirlo a Comunidad, en OPC, en la sección de Trabajos y Exámenes bajo la entrada “SgndExamen”.

#### OBSERVACIONES:

- En la figura 2, sólo se imprimen potencias eléctricas de índices existentes.
- Si en algún momento requiriera hacer algún “conteo a mano”, sin hacer cálculo, p.e. **mov totPot, 12;** puede hacerlo, pero sólo momentáneamente, si este conteo a mano queda dentro de la solución habrá un descuento.
- El procedimiento *ID* viene programado parcialmente; tiene que modificar su “textoID”.
- Dentro del programa viene dos call *DumpMem*, para probar parcialmente la funcionalidad del esqueleto inicial.



Nombre: Mónica Elizabeth Alba González Clave única: 160502 Calif: \_\_\_\_\_

Cada una de las preguntas: La 1 vale 3.0 pts., el 3 vale 2.0 pts y los restantes 2 y 4, valen 2.5 pts cada uno.  
**CONTESTAR CON TINTA.** Duración 1 hs 55 min o de 7-8:55 hs.

2.- Interprete el siguiente ejercicio.

.DATA

byteVal BYTE 1, 2, 3, 4 ; suponga un offset de 0 para byteVal

wordVal WORD 1008h, 2008h

WORD 3008h, 4008h

aString BYTE "ABCDEFGH", 0

dwordVal DWORD 12345678h, 34567890h

A) Interprete el siguiente código, indicando el valor final de ESI y que parte de la memoria es afectada.

MOV ESI, OFFSET dwordVal

ADD ESI, TYPE dwordVal

MOV DWORD PTR [ESI], 32198765h

B) Interprete el siguiente código, indicando los valores finales de EDI y EAX así como su respectivo significado.

MOV EAX, 0

MOV EDI, OFFSET wordVal

ADD EDI, SIZEOF wordVal

MOV AX, WORD PTR [EDI]

C) Interprete el siguiente código, indicando los valores finales de ESI y EDX así como su respectivo significado, como se modifica la memoria y lo impreso en la consola.

MOV ESI, 5

SUB BYTE PTR [aString+ESI], 3

MOV EDX, OFFSET aString

CALL WriteString

3.- Instrucciones y pedazos de programa

Conteste el siguiente inciso basándose en un ejemplo de uso. Indique la base en que está expresando los contenidos de los registros.

A) Suponga que no cuenta con la instrucción NEG, ¿Con cuál o cuáles instrucciones puede substituir la operación de la instrucción NEG?

B) ¿Cuáles son los valores finales de las banderas ZF y CF así como de AL? ¿Por qué?

MOV AL, 4

CMP AL, 5

C) ¿Cuál o cuáles son los valores de EAX y ECX, así como de algún otro registro involucrado?

MOV EAX, -48

CDQ

MOV ECX, 5

IDIV ECX

CONTINÚA EN LA SIGUIENTE PÁGINA. >>>

4.- Traduzca el siguiente programa de algún lenguaje de alto nivel a lenguaje ensamblador.

25  
2.5  
main(...) {

-----  
int uno, dos;  
int tres [] = {4, 5, 6, 7, 8, 9};  
int cuatro = 3;  
int cinco = 2;

uno = cuatro + 3 \* cinco;

if( uno <= 12 ) {  
    dos = -34;  
} else {  
    cinco = tres[1];  
}

dos = miProc(cuatro+20, cinco);

}

public static int miProc(int x, int y)  
{  
    System.out.println( (x-y) );  
    return x+y  
}

La implementación de las estructuras algorítmicas debe quedar muy claras, usando macro directivas.  
Los *int* considérellos como SDWORD.