

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Arquitectura de Computadoras y Ensambladores 1
Primer Semestre 2023
Ing. Otto René Escobar Leiva
Tutor Académico Sección A: Ronald Marín
Tutor Académico Sección B: Mario Pineda

Práctica 4

Descripción

Se le solicita el desarrollo de una aplicación de hojas de cálculo en Ensamblador. Dicho software permitirá manejar datos numéricos por medio de diversas operaciones matemáticas y lógicas, operando sobre datos puntuales o rangos. Se interactuará con el programa a través de una interfaz de línea de comandos por la cual se instruirá al programa sobre las acciones que debe realizar. El programa también tendrá la capacidad de generar salidas en varios formatos para diversos propósitos.

Diseño

En principio se definirán los elementos más importantes que deberán ser mostrados al usuario a la hora de ejecutar el programa. La cuadrícula, lo más importante, tendrá dimensiones limitadas; las columnas estarán marcadas de la letra A a la K, mientras que las filas estarán numeradas comenzando desde 1 a 23d. Sobre esta se llevarán a cabo las operaciones. La última línea de la pantalla es la línea de entrada, en donde será posible instruir, mediante comandos específicos, la operación del programa.

	A	B	C	D	E	F	G	H	I	J	K
01	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
02	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
03	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
04	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
05	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
06	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
07	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
08	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
09	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
10	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
11	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
12	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
13	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
15	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
16	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
17	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
18	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
19	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
20	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
21	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
22	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
24	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
:	-										

Celdas

El programa a desarrollar deberá entender el concepto de "celdas", las cuales están identificadas por un caracter alfabético y un número menor a 24. A la hora de ejecutar operaciones, el programa debe ser capaz de reconocer una referencia a una celda y hacerla efectiva.

Rangos

En este caso se entenderá a rango como una sucesión, vertical u horizontal, de celdas contiguas. Un rango tiene, una celda de inicio y de final. Más adelante se detallarán las operaciones que pueden hacerse sobre rangos.

Lenguaje de comandos

Para interactuar con el programa se definió un lenguaje de comandos que será el principal medio de interacción con la persona que utilice el programa. Las palabras o comandos reconocidos serán descritos más adelante.

Retorno de comandos

Según el comando, se indicará abajo, se generará un retorno. El valor de retorno podrá ser usado en los comandos que requieran una cantidad numérica como parámetro mediante el símbolo ``'``.

	A	B	C	D	E	F	G	H	I	J	K
00	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
01	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
02	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
03	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
04	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
05	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
06	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
07	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
08	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
09	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
10	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
11	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
12	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
13	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
15	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
16	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
17	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
18	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
19	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
20	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
21	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
22	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
24	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
: GUARDAR * EN C13_											

Comandos y operaciones

Operaciones sobre celdas

GUARDAR: GUARDAR [Número o celda] EN [Celda]

Con este comando es posible colocar un valor arbitrario en una celda. No se modifica el retorno.

SUMA: SUMA [Número o celda] Y [Número o celda]

Esta operación ejecutará una suma y el resultado de ésta será colocado en la variable de retorno.

RESTA: RESTA [Número o celda] Y [Número o celda]

Esta operación ejecutará una resta y el resultado de ésta será colocado en la variable de retorno. Si la resta realizada llegara a desbordarse, es necesario que se muestre una advertencia.

MULTIPLICACION: MULTIPLICACION [Número o celda] Y [Número o celda]

Se ejecutará una multiplicación de cantidades de 16 bits. El resultado será colocado en la variable de retorno. Si la operación llegara a generar un número que no puede ser representado con 5 dígitos decimales, será necesario indicar qué ha sucedido mediante una advertencia.

DIVIDIR: DIVIDIR [Número o celda] ENTRE [Número o celda]

El comando ejecutará una división, la parte entera del resultado será colocada en la variable de retorno. Debe tenerse en cuidado de indicar, por medio de una advertencia, cuando una división no sea posible de ejecutar.

POTENCIAR: POTENCIAR [Número o celda] A LA [Número o celda]

Esta operación permitirá ejecutar una potencia tomando como base el primer número y el número restante como exponente. El resultado deberá ser colocado en la variable de retorno. Debe verificarse que el exponente sea positivo pues el programa no considera potencias negativas. Y de igual forma, se debe revisar si el resultado generado por esta operación no puede ser representado en 5 dígitos. De ser así se debe mostrar una advertencia.

OLÓGICO: OLÓGICO [Número o celda] Y [Número o celda]

Este comando ejecutará un OR a nivel de bits con las cantidades brindadas como parámetros. El resultado de esta operación se colocará en el retorno.

YLÓGICO: YLÓGICO [Número o celda] Y [Número o celda]

Este comando ejecutará un AND a nivel de bits con las cantidades brindadas como parámetros. El resultado de esta operación se

colocará en el retorno.

OXLÓGICO: OXLÓGICO [Número o celda] Y [Número o celda]

Este comando ejecutará un XOR a nivel de bits con las cantidades brindadas como parámetros. El resultado de esta operación se colocará en el retorno.

NOLÓGICO: NOLÓGICO [Número o celda]

Este comando ejecutará un NOT a nivel de bits con las cantidad brindada como parámetro. El resultado de esta operación se colocará en el retorno.

Operaciones sobre rangos**LLENAR: LLENAR DESDE [Celda de inicio] HASTA [Celda de fin]**

Este comando deberá poder reconocer la orientación del rango especificado. Una vez reconocido se le solicitará a la persona que ingrese un valor numérico válido para cada celda del rango. De no ser válido lo ingresado se debe mostrar una advertencia. Una vez completado el ingreso de datos se deberá ver reflejado en la cuadrícula el cambio con los datos ingresados anteriormente colocados en su posición correcta. Esta operación no deberá modificar la variable de retorno.

PROMEDIO: PROMEDIO DESDE [Celda de inicio] HASTA [Celda de fin]

Este comando permitirá obtener la parte entera del promedio de un rango. El resultado se colocará en la variable de retorno para su posterior uso.

```
: LLENAR DESDE A15 HASTA A22
  VALOR PARA A15: 325
  VALOR PARA A16: 234223121
! Ingresar un numero valido.
  VALOR PARA A16: 123
  VALOR PARA A17: 42
  VALOR PARA A18: 242
  VALOR PARA A19: abcde
! Ingresar un numero valido.
  VALOR PARA A19: 220
  VALOR PARA A20: 884
  VALOR PARA A21: 552
  VALOR PARA A22: 7070
```

Ejemplo de uso de comando
LLENAR

MÍNIMO: MÍNIMO DESDE [Celda de inicio] HASTA [Celda de fin]

Por medio de este comando se podrá encontrar el valor mínimo en el rango especificado. Tal valor será el resultado que deberá ser colocado en el retorno.

MÁXIMO: MÁXIMO DESDE [Celda de inicio] HASTA [Celda de fin]

Por medio de este comando se podrá encontrar el valor máximo en el rango especificado. Tal valor será el resultado que deberá ser colocado en el retorno.

Entrada y Salida

IMPORTAR [Nombre del archivo] SEPARADO POR [COMA | TABULADOR]

Permitirá importar los datos numéricos de un archivo separado por comas o tabuladores. La primera línea del archivo tendrá los nombres de los encabezados, después de que se haya verificado que el archivo exista el programa pedirá la letra de la columna a la

que los datos correspondientes de cada encabezado irán. Teniendo un archivo como el siguiente:

```
Carne,Tarea 1,Parcial 1,Tarea 2
19001,90,70,85
19021,90,70,85
20901,90,70,85
19051,90,70,85
22001,90,70,85
19101,90,70,85
19731,90,70,85
```

El programa debe operar de la siguiente forma:

```
06 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
07 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
08 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
09 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
10 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
11 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
12 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
13 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
15 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
16 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
17 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
18 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
19 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
20 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
21 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
22 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
24 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
25 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
: IMPORTAR entrada.csv SEPARADO POR COMAS
Letra de la columna para "Carne": B
Letra de la columna para "Tarea 1": C
Letra de la columna para "Parcial 1": E
Letra de la columna para "Tarea 2": D
```

Los datos importados se colocarán en la columna indicada partiendo desde la primera fila hacia abajo.

Nota: se cuidará, en los archivos de entrada, que las líneas no sobrepasen los 256 caracteres.

EXPORTAR

Sintáxis: EXPORTAR [Número] DESDE [Letra de columna] HACIA [Nombre de archivo]

Este comando generará un archivo .htm que contendrá los datos tabulados. Específicamente se exportarán [Número] columnas desde

la columna [Letra de columna]. Debe colocarse la fecha y hora en que se exportaron los datos en alguna parte del archivo generado. Se debe tener cuidado de que la cantidad de columnas no sobrepase el número de columnas disponibles y otras verificaciones que se consideren pertinentes. Para cada columna se pedirá a quien use el programa el encabezado que esa columna tendrá en el archivo generado. Como lo muestra el ejemplo:

```

06 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
07 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
08 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
09 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
10 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
11 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
12 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
13 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
15 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
16 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
17 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
18 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
19 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
20 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
21 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
22 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
24 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
25 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
: EXPORTAR 3 DESDE D HACIA salida.htm
Encabezado para columna D:Codigo
Encabezado para columna E: Precio minorista
Encabezado para columna F: Precio mayorista_

```

Aclaraciones

- Los números a tratar dentro del programa son enteros con signo de 16 bits. Es decir cantidades representables con 6 caracteres, el primero sería el signo, o la ausencia de este en el caso de los positivos, y los 5 restantes el número en sí.
- Los archivos de entrada tendrán líneas de 256 caracteres o menos.
- Las advertencias en cuanto a inconsistencias aritméticas deben estar presentes para que se otorguen los puntos completos.
- Los estudiantes de carné PAR deben desarrollar la operación IMPORTAR tomando en cuenta archivos separados por COMAS.
- Los estudiantes de carné IMPAR deben desarrollar la operación IMPORTAR tomando en cuenta archivos separados por TABULADORES.
- Como mínimo, el programa, debe tener las operaciones IMPORTAR y GUARDAR desarrolladas.

- Los datos del estudiante deben ser impresos en pantalla de la misma forma, y con los mismos datos, que la Práctica 3.

Referencias

- Tabla ASCII: <https://elcodigoascii.com.ar>
- Libro de texto: Brey, Barry B. Microprocesadores Intel 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, and Pentium 4 Arquitectura, Programación e Interfaz. Pearson Prentice Hall. Séptima Edición.
- Libro sugerido: Assembly language for x86 processors. Irvine - Pearson - 2015

Entregables

- Manual técnico (hecho en Markdown)
- Manual de usuario (hecho en Markdown)
- Código funcional
- Link al repositorio de GitLab

Entrega

- La entrega, en UEDI, será el link al repositorio de gitlab
- El nombre del repositorio responde a la siguiente estructura:
 - o ACE1-
 - o Año
 - o Período
 - o Código curso
 - o Sección
 - o Número de carné
 - o PRAC4
- Ej. ACE1-221S0778A202301234PRAC4
- Realizar el último commit y hacer su entrega en UEDI antes de 23:59 horas del 27 de marzo.
- Se ejecutará un checkout hacia el último commit hecho antes de la fecha de entrega.
- El repositorio no debe contener ningún archivo EXE u OBJ, se

penalizará con el punteo del manual técnico.

- Se construirá el ejecutable al momento de calificación.
- Se debe agregar al auxiliar de su sección como developer del repositorio, si el auxiliar no fuese agregado o no le fuesen otorgados los permisos requeridos con antelación a la calificación no se podrá calificar.
 - o Sección A: @romasa000
 - o Sección B: @pinedaMario

Observaciones y restricciones

- La realización de la práctica es de forma individual
- Debido a la naturaleza del lenguaje ensamblador, solo se evaluarán salidas (resultados en pantalla o reportes) y no código.
- El código del programa debe ser estrictamente ensamblador, no se permite el uso de alguna librería.
- Se debe presentar el proyecto en Dosbox.
- Lenguaje ensamblador a utilizar MASM 6.11
- No está permitido el uso de estructuras de control if, if else, while, repeat, for.
- No está permitido el uso de STRUCT.
- Es requerido el uso de la directiva .RADIX 16
- Se debe entregar los manuales técnico y de usuario, de lo contrario se asumirá que el estudiante copió.
- Todas las acciones que amerite mostrar mensajes de error deberá mostrarlo, de lo contrario tendrá penalización en puntos.
- Los números de fecha y tiempo del reporte, en caso de ser menores a 10, deben incluir un cero antes del número (ej. 9 -> 09).
- El día de la calificación se harán preguntas, modificación de código sobre aspectos utilizados en la elaboración del proyecto, las cuales se considerarán en la nota final. Subir todos los archivos para que la práctica funcione así como los manuales para antes de 23:59 horas del lunes 27 de marzo.
- Copias parciales o totales tendrán una nota de 0 puntos y los

involucrados serán reportados a la Escuela de Ciencias y Sistemas

- Si el desarrollador no respeta, o cumple, las consideraciones especiales se asumirá que copió, o que no cumple con el mínimo requerido.