Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

CI-2692 – Laboratorio de Algoritmos y Estructuras II
Ene-Mar 2014

Proyecto 1 (15%)

Entrega: 19 de marzo de 2014 (jueves S.6) antes de las 11:59 PM en el espacio de su grupo en el Aula Virtual.
Los proyectos se realizan en parejas. Ambos miembros del equipo deben pertenecer a la misma sección.

Descripción General

Para este proyecto su grupo deberá crear un programa que permita realizar operaciones básicas con matrices de números enteros. Siguiendo los principios de encapsulamiento y modularidad, deberá implementar el **TAD Matriz**.

El mecanismo de entrada/salida de su programa será la lectura y escritura de archivos de texto. Cada archivo de entrada contendrá una secuencia de operaciones, entre las que se incluye la lectura de los valores de una matriz de otro archivo de texto. Los resultados de las operaciones deben escribirse en un archivo de salida.

Su programa se ejecutará mediante el comando especificado a continuación. Su módulo principal debe tener el mismo nombre.

```
> python3 matrix_calc.py <arch_instrucciones> <arch_salida>
```

Modelo Abstracto

A continuación se muestra una especificación incompleta del TAD Matriz. El primer paso del proyecto es completar la especificación formal y sustituir la explicación en lenguaje natural de cada operación por la pre- y post-condición. Puede usar notación matemática estándar para significar las operaciones entre matrices.

Especificación de TAD Matriz

Modelo de Representación

const *num_filas*, *num_columnas* : int
...???

Invariante de Representación

???

Operaciones

proc *crear* (**in** *p*, *q* : int ; **out** *m* : *Matriz*)
{ crea una matriz vacía de p filas y q columnas }
proc *obtener* (**in** *m* : *Matriz* , *fil*, *col* : int ; **out** *v* : int)
{ obtiene el valor contenido en la posición especificada de la matriz }
proc *colocar* (**in-out** *m* : *Matriz* , **in** *fil*, *col*, *v* : int)
{ coloca el valor especificado en la posición indicada de la matriz }
proc *sumar* (**in** *m1*, *m2* : *Matriz* , **out** *mf* : *Matriz*)
{ obtiene el resultado de sumar las dos matrices indicadas }

proc *mult_escalar* (**in** *m1* : *Matriz* , k : int, **out** *mf*: *Matriz*)
{ obtiene el resultado de multiplicar la matriz por el escalar provisto }
proc *trasponer* (**in** *m* : *Matriz* , **out** *mf*: *Matriz*)
{ obtiene la matriz traspuesta de la matriz provista }
proc *multiplicar* (**in** *m1*, *m2* : *Matriz* , **out** *mf*: *Matriz*)
{ obtiene el resultado de sumar las dos matrices indicadas }

Fin TAD

Formato de Instrucciones

El archivo de instrucciones estará compuesto de varias instrucciones separadas por saltos de línea. El archivo utilizará los strings M0, M1, M2… M9 para referirse a variables donde guardar los resultados de las operaciones.
Se opera bajo la convención de la primera fila y columna de una matriz tienen el índice cero.
Adicionalmente, al ejecutarse cada operación, el contenido de la variable de salida debe ser escrito al archivo de salida.
Las instrucciones válidas siguen a continuación. Las instrucciones con sintaxis o parámetros inválidos deben resultar en una excepción.

Sintaxis	Semántica	Salida
`READ <arch> <m>`	Leer el archivo *arch* y colocar su contenido en *m*	*m*
`NEW <p> <q> <m>`	*crear(p, q, m)*	*m*
`GET <m> <f> <c>`	*obtener(m, f, c, v)*	*v*
`SET <m> <f> <c> <v>`	*colocar(m, f, c, v)*	*m*
`ADD <m1> <m2> <m3>`	*sumar(m1, m2, m3)*	*m3*
`SCALAR <m> <k> <mf>`	*mult_escalar(m, k, mf)*	*mf*
`TRANSPOSE <m> <mf>`	*trasponer(m, mf)*	*mf*
`MULTIPLY <m1> <m2> <m3>`	*multiplicar(m1, m2, m3)*	*m3*

Ejemplo:

```
READ a.txt M0
NEW 2 2 M1
SET M1 0 1 -44
MULTIPLY M1 M0 M3
```

Formato de Datos

Los archivos que son parámetro de la instrucción READ tienen un formato definido. La primera línea contiene el número de filas y columnas de la matriz. Cada líneas siguientes indica el contenido de una fila de la matriz, estando los valores separados por uno o varios espacios en blanco.

Ejemplo:

```
2 3
5  7 14
-25 -1    7
```

Formato de Salida

Al imprimir matrices en el archivo de salida, éstas deberán estar delimitadas por secuencias de guiones iguales al ancho de la matriz. Los valores deberán estar alineados y justificados a la derecha (el número de espacios necesarios para la justificación viene dado por el máximo y mínimo valor contenido en la matriz). Si lo desea, puede hacer uso de código ya escrito y/o librerias predefinidas de *Python* para este fin.

En el caso de la instrucción GET, puede imprimirse el valor de salida en una sola línea, sin formato adicional.

Ejemplo:

```
-----------
  5   7  14
-25  -1   7
-----------
-----
 0 -4
 1  2
-----
-42
```

Requerimientos de la Entrega

1. La entrega consiste en los siguientes artefactos. Se recomienda elaborarlos en el orden propuesto.
   a) Especificación completa con el modelo abstracto del TAD Matriz. Puede ser escrita a mano y escaneada, pero en todo caso debe ser perfectamente legible.
   b) Especificación con el modelo concreto de su escogencia del TAD Matriz. Igual que el punto anterior.
   c) Implementación del proyecto.
   d) Archivo `readme.txt` con instrucciones breves para correr el proyecto, documentación de problemas y/o errores y cualquier otro comentario que considere pertinente.

2. Se evaluará positivamente la limpieza y buena estructura del código. En particular procure separar las operaciones del TAD de la lógica de entrada-salida de su programa. Se recomienda que la implementación del TAD Matriz esté en su propio archivo .py, distinto del módulo principal `matrix_calc.py`.

3. El código debe estar adecuadamente documentado, lo cual incluye nombre de variables y métodos auto-explicativos en la medida de lo posible. Recuerde que el propósito de la documentación no es repetir información ya expresa en el código.

4. Cada integrante del grupo deberá imprimir, firmar y entregar a su profesor la *"Declaración de Autenticidad para Entregas"*, cuya plantilla se encuentra en el Aula Virtual del curso.

5. La entrega del proyecto será directamente en el espacio de su grupo en el Aula Virtual. Deposite un archivo comprimido que contenga los archivos de código fuente, así como otros que sean necesarios para la ejecución de la aplicación. El nombre del archivo debe ser `Proy1-XX.tar.gz`, donde XX es el número de su grupo.

6. Todos sus archivos de código deben tener un encabezado con los nombres y números de carnet de ***ambos*** miembros del equipo.

7. La entrega del proyecto debe realizarse **ANTES** de la fecha y hora indicadas al inicio del

enunciado. Se recomienda realizar la entrega con antelación y considerar imprevistos como fallas de conectividad.

Se proveerá una muestra de archivos de entrada y la salida esperada junto con este enunciado. Igualmente se adjuntará código con un ejemplo básico de lectura y escritura de archivos.

El no cumplimento de los requerimientos podrá resultar en el rechazo de su entrega.

Preguntas y consultas

Se recomienda hacer preguntas con la mayor antelación posible. A fin de realizar las consultas de manera pública y persistente, coloque sus preguntas en el foro del Aula Virtual.

Saúl González D. (esaulgd) 2014.03.06 v1.1