

UNTREF

Algoritmos y programación 3

Trabajo práctico Final: TDA generación de archivos JSON

1ero cuatrimestre, 2018

(Trabajo Grupal)

Alumnos:

Nro	Nombre	Legajo	Mail
1	Maximiliano Andreoli	39439	andreolixam@gmail.com
2	Andres Ramos	39412	andresgabriel.92@gmail.com
3	Franco De Alesandro	38583	francodealesandro@@gmail.com

Fechas de entrega programadas

- 1er Entrega : 5/5/2018
- 2da Entrega : 19/05/2018
- 3er Entrega : 9/6/2018
- 4ta y última Entrega : 30/06/2018

Nota Final:

Introducción

Objetivo del trabajo

Aplicar los conceptos enseñados en la materia a la resolución de un problema así como también la aplicación de buenas prácticas, trabajando en forma grupal y utilizando un lenguaje de tipado estático de bajo nivel (Lenguaje C)

Consigna general

Desarrollar un conjunto de TDAs que permitan operar con un conjunto de JSON (JavaScript object notaton)) Una vez creado este conjunto de datos utlizarlo en la generación de un archivo JSON) La aplicación deberá ser acompañada con la apropiada documentación de diseño) En la siguiente sección se describe la aplicación a desarrollar)

Descripcción de la aplicación a desarrollar

Definir un conjunto de TDAs que permitan generar el siguiente archivo JSON:

```
{
  "size": "0 bytes",
  "hash": "37eb1ba1849d4b0fb0b28caf7ef3af52",
  "bytes": 0,
  "thumb_exists": false,
  "rev": "714f029684fe",
  "modified": "Wed, 27 Apr 2011 22:18:51 +0000",
  "path": "/Photos",
  "is_dir": true,
  "icon": "folder",
  "root": "dropbox",
  "contents": [
    {
      "size": "2.3 MB",
      "rev": "38af1b183490",
      "thumb_exists": true,
      "bytes": 2453963,
      "modified": "Mon, 07 Apr 2014 23:13:16 +0000",
      "client_mtime": "Thu, 29 Aug 2013 01:12:02 +0000",
      "path": "/Photos/flower.jpg",
      "photo_info": {
        "lat_long": [
          37.772566666666666,
          -
122.45934166666667
        ],
        "time_taken": "Wed, 28 Aug 2013 18:12:02 +0000"
      },
      "is_dir": false,
      "icon": "page_white_picture",
      "root": "dropbox",
      "mime_type": "image/jpeg",
      "revision": 14511
    }
  ],
  "revision": 29007
}
```

Consignas para el programa principal:

Luego de definir apropiadamente las estructuras crear el programa principal que generará este archivo)
Dicho programa podrá recibir como parámetro:

- **Ningún parámetro:** En cuyo caso el JSON se imprimirá en la consola (standard output)
- **Parámetro: -f <archivo_destino>** En cuyo caso el JSON generado deberá escribirse en un archivo llamado *archivo_destino*)

La API para manejar JSON deberá proveer funcionalidad para realizar las siguientes tareas:

1. Crear un nuevo *nodo JSON* (en adelante *nJson*) asignandoles nombre y valor a *nJSON*.
- 2) Cambiarle el contenido* al *nJson* 3)
Agregar nuevo *nJson* a un *nJson* existente

**Respecto a los contenidos posibles:*

- 1) *nJson*)
 - 2) Cualquier tpo de dato punto flotante)
 - 3) Cualquier tpo de dato entero)
 - 4) Strings)
 - 5) Booleans)
 - 6) Arrays
- a) De cualquier tpo entre 1-5 (solo un tpo por vez))

NOTA:

Los contenidos del *nJson* en ocasiones pueden ser Números enteros / Flotantes / Booleanos / Strings / etc) En estos casos el usuario de la API deberá poder operar con los tpos de datos nativos provistos por C que mejor se adapten al dato en cuestión)

Pautas de calidad al momento de resolver el TP

- Los atributos del TDA deberán accederse mediante funciones provistas por la API **exclusivamente** (Applicaton Programming Interface))
- Dentro de las funciones que operan sobre el TDA/s evitar los llamados a funciones que impriman en las salidas estándar de errores o en la salida estándar (print, fprintf, etc))
- Cada uno de los TDAs definidos deberán estar apropiadamente documentados, tanto sus campos como todas sus primitivas)
- Elegir un standard de codificación y seguirlo a lo largo del proyecto)
- Toda la memoria reservada dentro del TDA será responsabilidad de la API, es decir, si el TDA reserva memoria deberá proveer las herramientas necesarias para liberarla apropiadamente)
- Si necesita utilizar números mágicos, utilice #define y documéntelos apropiadamente)
- Los arrays utilizados dentro del TP deberán poder ser de una longitud indefinida (sin restricciones en su cantidad de elementos))
- La estructura del proyecto deberá ser:
 - o Archivo main.c con el ejemplo de uso de el/los TDA/s
 - o Archivo <declaración del tda>.h
 - o Archivo <definición del tda>.c
- El proyecto deberá ser implementado utilizando la interfase Eclipse CDT)

Formas de entrega

Habr  4 entregas formales) Las mismas tendr n una calificaci n de APROBADO o NO APROBADO en el momento de la entrega. Cada uno de las entregas deber  contar con:

- Estructura de proyecto tal como lo crea el entorno Eclipse CDT.
- Este informe en el formato que lo distribuy  el docente.
- El informe debe contar con los datos de los integrantes del grupo.
- Cada uno de los requerimientos de la entrega deber n estar agrupados en un contexto (scope) independiente.
- La entrega se realiza digitalmente v a mail. Nombrar el archivo como TP_<apellido_0>_<apellido_1>_..._<apellido_n>.zip

1er Entrega:

- Estructura completa del proyecto)
- Creaci n del main.c y la lectura de par metros que provean la posibilidad de ejecutar la aplicaci n en los dos modos)

2da Entrega:

Tener la posibilidad de:

- 1) Entrega 1,
- 2) Crear un nJson
- 3) Poder asignarle cualquier contenido (salvo array) para todos los tipos primitivos de C) Agregar la posibilidad de poder imprimir el nJson)

3ra Entrega:

Tener la posibilidad de:

- 1) Entrega 2,
- 2) Cambiarle el contenido* al nJson
- 3) Agregar nuevo nJson a un nJson existente
- 4) Poder asignar como contenido de un njson un array
- 5) Completar la salida por pantalla)
- 6) Crear otro JSON y mostrarlo en la salida utilizando el par metro -C)

4ta y  ltima Entrega:

Trabajo Pr ctico completo, funcionando, con el informe y cumpliendo todas las normas de calidad)

IMPORTANTE:

Si el TP no se puede ejecutar por cualquier tipo de error (acceso inv lido de memoria, goteo de memoria, etc) el TP estar  **desaprobado**.

Fechas de entrega programadas

- 1er Entrega : 5/5/2018
- 2da Entrega : 19/05/2018
- 3er Entrega : 9/6/2018
- 4ta y  ltima Entrega : 30/06/2018

Checklist de corrección

Esta sección es para uso exclusivo de los docentes, por favor no modificar)

Carpeta

Generalidades

- ¿Son correctos los supuestos y extensiones?
- ¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tpograaa

uniforme) **Modelo**

- ¿Está completo? ¿Contempla la totalidad del problema?
- ¿Respeto encapsulamiento?
- ¿Pierde memoria?
- ¿Cumple con las buenas prácticcas?

Código

Generalidades

- ¿Respeto estándares de codifcación?
- ¿Está correctamente documentado?
- ¿Defne magic numbers? ¿Están documentados? □ ¿Respeto estructura del proyecto?