

Calculador de desempeño académico

95.11/75.02 Algoritmos y Programación I

6 de septiembre de 2017

1. Objetivo del TP

El objetivo del presente trabajo consiste en la realización de un conjunto de aplicaciones en modo consola, escritos en lenguaje ANSI-C89, que permitan implementar un sistema de cálculo de métricas sobre el desempeño académico del usuario.

2. Alcance del TP

Mediante el presente TP se busca que el estudiante adquiera y aplique conocimientos sobre los siguientes temas:

- Programas en modo consola
- Directivas al preprocesador C
- Juego de caracteres ASCII
- Tipos enumerativos
- Control de flujo
- Salida de datos con formato
- Funciones
- Modularización
- Arreglos
- Cadenas de caracteres

Fecha de entrega: 20 de septiembre de 2017

3. Introducción

Es común entre los alumnos llevar una cierta medida de cómo el mismo se va desempeñando a lo largo de la carrera. Para esta tarea, se pide desarrollar una aplicación que, cumpliendo con las especificaciones de la sección 4, pueda mostrar una serie de datos según la información provista por el alumno.

4. Especificaciones

El programa a desarrollar debe ser interactivo. Debe obtener sus datos a través del flujo de entrada estándar, `stdin`. Deberá consultar al usuario por los siguientes datos: nombre, apellido, padrón, carrera y asignaturas. La cantidad de materias es desconocida y serán ingresadas de a una, con su nombre y nota. Además, debe ser posible compilarlo en diferentes idiomas.

Ante cada error en el ingreso de los datos, el programa debe indicar al usuario, con un mensaje *claro*, la falla.

4.1. Menú principal

Para la toma de datos, el programa se valdrá de un menú, similar al siguiente ejemplo:

```
Elija qué desea hacer:
  R) Modificar registro
      personal
  A) Modificar asignatura
  M) Calcular métrica
      individual (opcional)
5  !) Finalizar
   0) Salir
?
```

Ejemplo 1: Español

```
Velg hva du skal gjøre:
  P) Endre
      personopplysninger
  F) Endre fag
  B) Beregn individuelle
      beregninger
5  !) Slutt
   0) Avslutt
?
```

Ejemplo 2: Idioma 2

Cada uno de los ítems del menú puede o no tener un submenú en el que debe actuar. Nótese que los caracteres de selección pueden ser variados: alfabéticos, símbolos y/o numéricos. A continuación se detallan las operaciones de cada uno de estos ítems.

4.2. Submenú de registro personal

En este menú se pide información para completar lo que llamaremos “registro personal”. Estos datos abarcan: nombre, apellido, padrón y carrera.

La selección del ítem a completar depende de un nuevo menú, a saber:

```
Elija qué desea modificar:
  A) Apellido y nombre
  #) Padron
  C) Carrera
```

```
5 | 0) Volver
  | ?
```

Al seleccionar una opción, se mostrará el dato cargado, se solicitará el nuevo dato, y se registrará el mismo, sin pedido de confirmación.

4.3. Submenú de asignaturas

En este menú se presenta la información correspondiente a las asignaturas y se registran cambios y nuevos datos. Un ejemplo de ejecución de este menú es:

```
Usted no tiene asignaturas cargadas.
+) Ingresar nueva asignatura
0) Volver
?
```

ó bien

```
Opciones:
 1) AMII (9)
 2) Química (8)
 3) Inglés (10)
5 4) Probabilidad y estadística (9)
 5) Sociología (10)
+) Ingresar nueva asignatura
-) Eliminar una asignatura (opcional)
10 0) Volver
  ?
```

Nótese la particularidad que tiene este menú de modificarse según el estado de ejecución del programa.

4.4. Submenú de métricas (opcional)

En este menú se da la opción al usuario de calcular alguna métrica con el estado actual del programa y presenta un menú similar al siguiente:

```
Métricas:
 P) Promedio
 M) Máximo
 m) Mínimo
5 #) Cantidad de materias
 2) Aplazos
 0) Volver
 ?
```

Seleccionada una opción, el programa indica al usuario dicho resultado.

4.5. Submenú de finalización

Este submenú no mostrará un nuevo menú. En el mismo se computarán todos las métricas y se imprimirá un mensaje, por una salida estándar diferente a la utilizada en los menús. El mensaje debe respetar el siguiente formato:

```
Nombre y Apellido , Padrón , Carrera , Materias , Promedio , Aplazos
```

Todo el mensaje contenido en una única línea. Luego se volverá al inicio de la aplicación, con todos los datos eliminados.

4.6. Submenú "Salir"

Termina el ciclo de ejecución, sin mostrar mensajes.

5. Opcionales

Se consideran puntos opcionales, la posibilidad de eliminar asignaturas ya cargadas de la lista, y la posibilidad de computar cada una de las diferentes métricas a pedido del usuario.

Se puede agregar, como opcional, que el cada uno de los caracteres que se utilizan en las selecciones no varíe con el idioma. De hacer este opcional, se considerará válido si la implementación es correcta, ya que no es correcto mantenerlos a través de hardcodes o modularizaciones pobres.

6. Ejemplo de ejecución

```
$ ./mi_aplicacion
Bienvenidx
Elija qué desea hacer:
5   R) Modificar registro personal
   A) Modificar asignatura
   M) Calcular métrica individual
   !) Finalizar
   O) Salir
? R
10  Elija qué desea modificar:
   A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
15  ? N
   No tiene nombre cargado.
   ¿Apellido y nombre? Fulanito Cosme
   Elija qué desea modificar:
20  A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
   ? #
   ¿Padron? 123456
25  Elija qué desea modificar:
   A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
30  ? C
   ¿Codigo de la carrera? 7
   Elija qué desea modificar:
35  A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
   ? O
   Elija qué desea hacer:
```

```
40      R) Modificar registro personal
      A) Modificar asignatura
      M) Calcular métrica individual
      !) Finalizar
      0) Salir
? A
45 Usted no tiene asignaturas cargadas.
  Opciones:
      +) Ingresar nueva asignatura
      0) Volver
? +
50 ¿Asignatura (Nota Nombre)? 9 AMII
  Opciones:
      1) AMII (9)
      +) Ingresar nueva asignatura
      0) Volver
55 ? 0
  Elija qué desea hacer:
      R) Modificar registro personal
      A) Modificar asignatura
      M) Calcular métrica individual
60      !) Finalizar
      0) Salir
? !
  Fulanito Cosme,123456,7,1,9,0
  Elija qué desea hacer:
65      R) Modificar registro personal
      A) Modificar asignatura
      M) Calcular métrica individual
      !) Finalizar
      0) Salir
70 ? 0
$
```

7. Comentarios

Si bien no es necesario utilizar variables globales en este trabajo, con su debida **justificación**, *algunas* pueden ser aceptadas.

Es recomendable tener un diccionario construido con las carreras como cadenas de caracteres y un tipo enumerado; lo mismo que los mensajes de error y el enumerando `status_t`. Este enumerando, en español representa una *condición*.

El nombre del enumerando que representa *estado* en el que se encuentra el programa (menú principal, algún submenú, error, inicio, reinicio, etc.) en inglés es `state_t`.

8. Restricciones

La realización del trabajo se encuentra sujeta a las siguientes restricciones:

- Debe realizarse en grupos de **3 (tres)** integrantes.
- No está permitida la utilización de memoria dinámica.

- No está permitida la utilización de funciones específicas para manejo de archivos. Como funciones de E/S sólo pueden ser utilizadas las funciones de biblioteca `fgetc()`, `getchar()`, `scanf()`, `printf()`, `fprintf()`, `putc()`, `fputc()`, `puts()`, `fputs()` o similares.
- No está permitida la utilización de `gets()`¹, `fflush(stdin)`², la biblioteca `conio.h`³ (`#include <conio.h>`), etc.
- Debe recurrirse a la utilización de funciones mediante una adecuada parametrización.
- No está permitido en absoluto tener hard-codings:

```

1 ...
2 char c;
3 ...
4 if (c == 'R')                               /* ; ;hard-coded!!*/
5     printf("%s", "xxxxxxxx");               /* ; ;hard-coded!!*/
6 else if (c == 'A')                           /* ; ;hard-coded!!*/
7     printf("%s", "yyyyyyyy");              /* ; ;hard-coded!!*/
8 ...

```

sino que debe recurrirse al uso de ETIQUETAS, CONSTANTES SIMBÓLICAS, MACROS, etc.

Los ejemplos no son exhaustivos, sino que existen otros hard-codings y tampoco son aceptados.

- Hay ciertas cuestiones que no han sido especificadas intencionalmente en este Requerimiento, para darle al/la desarrollador/a la libertad de elegir implementaciones que, según su criterio, resulten más convenientes en determinadas situaciones. Por lo tanto, se debe explicitar cada una de las decisiones adoptadas, y el o los fundamentos considerados para las mismas.

9. Entrega del Trabajo Práctico

La fecha de entrega del trabajo práctico es: 20 de septiembre de 2017 o antes.

No se requiere entrega en papel. Deberá realizarse una entrega digital, a través del campus de la materia, de un único archivo cuyo nombre debe seguir el siguiente formato:

YYYYMMDD_apellido1-apellido2-apellido3_entrega-N.tar.gz

donde YYYY es el año (2017), MM el mes y DD el día en que uno de los integrantes sube el archivo, apellido-1a3 son los apellidos de los integrantes ordenados alfabéticamente, entrega-N indica el número de vez que se envía el trabajo (entrega-1, entrega-2, etc.), y `.tar.gz` es la extensión, que no necesariamente es `.tar.gz`.

¹obsoleta en C99 [3], eliminada en C11 [4] por fallas de seguridad en su uso.

²comportamiento indefinido para flujos de entrada ([3],[4]). Definida en estándar POSIX.

³biblioteca no estándar, con diferentes implementaciones y licencias, y no siempre disponible.

El archivo comprimido debe contener los siguientes elementos:

- La correspondiente documentación de desarrollo del TP (en formato pdf), siguiendo la numeración siguiente, incluyendo:
 1. Carátula del TP. Incluir una dirección de correo electrónico.
 2. Enunciado del TP.
 3. Estructura funcional de los programas desarrollados.
 4. Explicación de cada una de las alternativas consideradas y las estrategias adoptadas.
 5. Resultados de la ejecución (corridas) de los programas, captura de las pantallas, bajo condiciones normales e inesperadas de entrada.
 6. Reseña sobre los problemas encontrados en el desarrollo de los programas y las soluciones implementadas para subsanarlos.
 7. Bibliografía (ver aparte).
 8. Indicaciones sobre la compilación de lo entregado para generar la aplicación.

NOTA: Si la compilación del código fuente presenta mensajes de aviso (warning), notas o errores, los mismos deben ser comentados en un apartado del informe.

NOTA: El Informe deberá ser redactado en *correcto* idioma castellano.

- Códigos fuentes en formato de texto plano (.c y .h), *debidamente documentados*.

NOTA: Todos los integrantes del grupo deben subir el *mismo* archivo.

NOTA: Se debe generar y subir un único archivo (comprimido) con todos los elementos de la entrega digital. **NO usar RAR**. La compresión RAR no es un formato libre, en tanto sí se puede utilizar *ZIP*, *GUNZIP*, u otros (soportados, por ejemplo, por la aplicación de archivo *TAR*).

Si no se presenta cada uno de estos ítems, será rechazado el TP.

10. Bibliografía

Debe incluirse la referencia a toda bibliografía consultada para la realización del presente TP: libros, artículos, URLs, etc., citando:

- Denominación completa del material (Título, Autores, Edición, Volumen, etc.).
- Código ISBN del libro (opcional: código interbibliotecario).
- URL del sitio consultado. No poner *Wikipedia.org* o *stackexchange.com*, sino que debe incluirse un enlace al artículo, hilo, etc. consultado.

Utilizando *L^AT_EX*, la inclusión de citas/referencias es trivial. Los editores de texto gráficos de las suites de ofimática, como LibreOffice Write o MS Word, admiten plugins que facilitan la inclusión.

Ejemplo de referencias

- [1] B.W. Kernighan y D.M. Ritchie. *The C Programming Language*. 2.^a ed. Prentice-Hall software series. Prentice Hall, 1988. ISBN: 9780131103627.
- [2] P. Deitel y H. Deitel. *C How to Program*. 7.^a ed. Pearson Education, 2012. ISBN: 9780133061567.
- [3] ISO/IEC. *Programming Languages – C*. ISO/IEC 9899:1999(E). ANSI, dic. de 1999, págs. 270-271.
- [4] ISO/IEC. *Programming Languages – C*. INCITS/ISO/IEC 9899:2011. INCITS/ISO/IEC, 2012, pág. 305.