Laboratorio de Programación de Computadoras: Examen Final

Nombre:_____ Cal.:_____

27 de Junio de 2019

1. Hacer un programa que recibe como parámetros de entrada: a) Tú nombre; b) Matrícula; y c) Edad. El programa debe escribir en un archivo los datos que recibe como parámetro (**1 punto**). Además el programa debe tener una función que imprime en el archivo la secuencia de números que van desde -Edad hasta Edad (**1.5 puntos**).
   El ejemplo de ejecución del programa es: `./a.out FERNANDO 8900313g 40`
   El contenido del archivo que se genera desde dentro del programa es:

```
FERNANDO
8900313g
-40 -39 -38 -37 -36 -35 -34 -33 -32 -31 -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19
-18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35   36 37 38 39 40
```

2. Escriba un programa en lenguaje C que haga lo siguiente:

   (a) (**0.5 puntos**) Recibe el nombre de un archivo en la línea de comandos, es decir la función `main` recibe argumentos de entrada. Imprimir en la pantalla el nombre del archivo que se recibe.

   (b) (**1 punto**) Con el nombre de archivo que se recibe, debe abrirse un archivo en modo lectura y leer todos los valores enteros del archivo. Imprimir en la pantalla los valores enteros que se leen.

   (c) (**1 punto**) Crear un nuevo archivo dentro del programa, en donde debe escribir la equivalencia a caracter de cada valor entero que se ha leído en el paso anterior, al finalizar cierre los archivos abiertos. Descargar del sitio: `https://lc.fie.umich.mx/~moises/apuntes/Lab_Programaci%c3%b3n_Computadoras/EXAMENES/` un ejemplo del archivo `valores.txt`.

   Ejemplo de ejecución del programa: `./a.out valores.txt`
   El contenido del archivo de salida para el archivo valores.txt es:

```
  Para declarar una variable de tipo apuntador a archivo:
#include <stdlib.h>
FILE *fp;
Ejemplo de apertura de un archivo:
fp = fopen("datos.txt","w");
La función fflush se usa para fÍsicamente escribir (bajar) en el archivo cualquier data
almacenada en el "buffer" intermedio. Para hacer los programas mas eficientes y reducir
las llamadas aL sistema, los archivos mantienen un buffer (memoria) intermedia, una vez
que el buffer esta lleno, los datos se escriben en el archivo.
```

3. Hacer un programa que define la estructura de datos carita y las siguientes acciones para crear una imagen en el formato SVG:

Figure 1: Ejemplo del dibujado de dos caritas en un archivo svg.

(a) (**1 punto**) Declarar una variable del tipo de dato carita y que el usuario proporcione los datos definidos dentro de la estructura carita, es decir, que se le asignen valores a las variables dentro del tipo de dato definido.

(b) (**0.5 puntos**) Hacer que en forma aleatoria se seleccione el color de relleno y borde de la carita.

(c) (**1 punto**) Una función que recibe como parámetro el tipo de dato carita y un nombre de archivo de imágenes en el formato svg. La función dibuja dos caritas en el archivo imagen svg. La primera con los datos que introduce el usuario y la segunda mediante un valor de desplazamiento en forma aleatoria en el eje X, ver la Figura 1.

4. (**2.5 puntos**) Hacer un programa que lee dos arreglos de números flotantes, el usuario indica el tamaño de los arreglos. En un tercer arreglo se debe almacenar el valor promedio de cada una de las posiciones de los primeros dos arreglos. Ver el ejemplo para arreglos de tamaño 10. Almacenar en un archivo con el nombre que usted guste el tercer arreglo.

Arreglo 1:

-3.5	4.12	1.5	6.5	8.4	-1.25	2.5	-3.8	4.4	5.6

Arreglo 2:

-1.5	1.78	3.5	1.5	1.6	-1.75	-2.5	-1.2	14.6	5.4

Arreglo 3: Almacena los valores promedio de los arreglos 1 y 2

-2.000	2.950	2.500	4.000	5.000	-1.500	0.000	-2.500	9.500	5.500