Maestría en Ciencias de la Computación
Proceso de admisión 2019
Centro de Investigación en Matemáticas (CIMAT), A.C.
Examen de programación
(Tiempo: 4 horas)

Nombre: _____

Fecha: _____

Instrucciones:

- Hacer **dos grupos de hojas con sus respuestas**, uno que contenga las respuestas a las preguntas 1 a 3, otro para las respuestas 4 a 6.
- Si se pide código, escribirlo lo más claramente posible. Especificar el lenguaje/pseudo-código utilizado.
- Si requiere funciones estándar como determinar el máximo o mínimo de dos valores (max(a, b), min(a,b)), ordenar (sort), determinar el tamaño de una cadena de caracteres (strlen), obtener el valor absoluto (abs), raíz cuadrada (sqrt) o logaritmos (log) puede utilizarlas sin implementarlas. Será necesario que implemente cualquier función diferente a las mencionadas anteriormente.

Problema 1 [1 puntos]

Dado un entero N, escribir una función **int reverseBase10(int N)** que invierta el orden de los digítos de N cuando ese número está expresado en base 10.

Ejemplo: si N = 12345, la función debería regresar el número 54321.

Problema 2 [1.5 puntos]

Escribir una función **int getSecondMax(int data[N])** que reciba de entrada un arreglo con N elementos distintos y que regrese el segundo máximo elemento del arreglo.

Ejemplo: al correr esta función con el siguiente arreglo como entrada

$data = \{1, 20, -5, 345, -3, 2\}$

se debería regresar 20, que es el segundo máximo elemento del arreglo.

Problema 3 [1.5 puntos]

Se suele representar una imagen numerícamente por una matriz de valores $\mathbf{I}$ tal que $\mathbf{I}[x][y]$ indique el valor de nivel de gris en la posición indicada por los índices x y y. Se supondrá aquí que estos valores son enteros positivos encodificados en 8 bits (entre 0 y 255).

Escribir una función que tome de entrada una matriz-imagen $\mathbf{I}$ cuadrada de tamaño $\mathbf{m}\times\mathbf{m}$ y que le apliqué una rotación de 90 grados hacia la derecha (como cuando uno rota la imagen en su celular).

Ejemplo: ante la ejecución de la función con la siguiente imagen

$$\mathbf{I} = \begin{bmatrix} 10 & 50 & 70 & 0 \\ 200 & 200 & 170 & 200 \\ 0 & 45 & 120 & 156 \\ 2 & 78 & 89 & 20 \end{bmatrix}$$

la función debe devolver

$$\mathbf{I}' = \begin{bmatrix} 2 & 0 & 200 & 10 \\ 78 & 45 & 200 & 50 \\ 89 & 120 & 170 & 70 \\ 20 & 156 & 200 & 0 \end{bmatrix}.$$

Problema 4 [2 puntos]

Matching$^{\text{TM}}$ es una agencia matrimonial con métodos muy raros. Cada sábado, proceden a organizar citas entre sus adherentes (hombres H y mujeres M) de la siguiente manera: forman una linea de N soltero(a)s, hombres y mujeres, en el orden que acudieron a la agencia, y autorizan que un hombre y una mujer tomen cita si están a una distancia inferior o igual a K en la linea.

Escribir una función que tome de entrada el arreglo A y el entero K y determine cuál es el número máximo de citas que se pueden organizar.

Ejemplo: ante la ejecución de la función con el siguiente arreglo

$A = \{H, H, H, F, H, H, F, F\}$

y con $K = 1$ la función debe devolver 2.

Problema 5 [2 puntos]

La función factorial está definida para $n \in \mathbb{N}$ por

$$\begin{aligned} 0! &= 1 \\ n! &= 1 \times 2 \times ... \times n \text{ si } n > 0. \end{aligned}$$

Por otro lado, el desarrollo en serie de Taylor de la funció $\sin(x)$ alrededor de 0 y al orden $2n+1$ está dado por

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + o(x^{2n+1})$$

donde $o(x^{2n+1})$ son términos de orden superior a $2n+1$, despreciados en la aproximación.

1. Al calcular de la manera más ingenua posible las factoriales involucradas en la approximación por serie de Taylor al orden $2n+1$ de $\sin(x)$, ¿cuántas multiplicaciones de enteros naturales tendremos que hacer en función de n?
2. Al calcular de la manera más ingenua posible las potencias de x (x, x^3, $x^5 \ldots$) requeridas para determinar la approximación por serie de Taylor al orden $2n+1$ de $\sin(x)$, ¿cuántas multiplicaciones por x de tipo $x \times x \times ...$ tendremos que hacer en función de n?
3. Desarrolle una función que tome como entrada un x y un $n \geq 0$ y evalúe la aproximación por serie de Taylor de $\sin(x)$ al orden $2n+1$, donde se buscará utilizar el mínimo número posible de multiplicaciones para los cálculos de factoriales y de potencias de x.
   Ejemplo: para $x = 0{,}1$ y $n = 2$, se deberá evaluar $\frac{0{,}1}{1!} - \frac{0{,}001}{3!} + \frac{0{,}00001}{5!}$.

Problema 6 [2 puntos]

Escribir una función que determine si un punto del plano está adentro o afuera de un polígono con N vértices. Esta función **bool testPointInPolygon(int pol_ver_x[N], int pol_ver_y[N], int point_x, int point_y)** recibe los N vértices consecutivos del polígono, a través de los arreglos **int pol_ver_x[N]** y **int pol_ver_y[N]**. Cada uno de estos arreglos contiene las coordenadas x y las coordenadas y de los vértices, respectivamente. Suponemos que las únicas intersecciones que existen entre las aristas del polígono son sus vértices. El punto que queremos probar es (point_x, point_y) y suponemos que nunca estará ubicado sobre la frontera del polígono.

Ejemplo: si N = 3, pol_ver_x = {0, 1, 2}, pol_ver_y = {0, 2, 0}, point_x = 1 y point_y = 1, entonces el método debería regresar true. La figura siguiente muestra para este caso la configuraciónón del polígono así como la ubicación del punto que probar.