

Examen de Programación para Ingreso a la
Maestría en Ciencias de la Computación 2015
CIMAT, A.C.
(Tiempo: 4 horas)

Nombre: _____

Fecha: _____

Instrucciones:

- Hacer **dos grupos de hojas con sus respuestas**, uno que contenga las respuestas a las preguntas 1 a 4, otro para respuestas a las preguntas 5 en adelante.
- Si se pide código, escribirlo lo más claramente posible. Especificar el lenguaje/pseudo-código utilizado.
- En todos los problemas, los índices utilizados en vectores o matrices empiezan con el valor 0.

Problema 1 [1.0 punto]

Se tiene un arreglo de caracteres **C** y se sabe que el último elemento de **C** es el único que contiene un cero. Dar el código para contar las ocurrencias aisladas del carácter 'a' (es decir, cuando los caracteres vecinos a una 'a' no sean 'a's).

Ejemplo: ante la ejecución de la función con la siguiente cadena de entrada:

$\{a, d, a, a, f, a, g, r, r, a, 0\}$,

se debe devolver 3.

Problema 2 [1.5 punto]

Implemente una función **void SumDiagonals(int data[N][M], int result[])** que calcule la suma de cada una de las diagonales cuya orientación es hacia la derecha y abajo, en la matriz data, y almacene los resultados en el array result.

Ejemplo: ante la ejecución de la función con la siguiente matriz data:

$$\begin{bmatrix} 10 & 26 \\ 12 & 20 \\ -5 & 4 \end{bmatrix}$$

la función debe almacenar en result el siguiente contenido: {26, 30, 16, -5}.

Problema 3 [1.0 punto]

En el código siguiente, ¿cuál es el valor de *i* al final de la ejecución?

```

int x=1;
int i=1;
while (x<=1000) {
    x = pow(2,x);
    i = i +1;
}

```

Problema 4 [1.5 punto]

Sean $\mathbf{X} = \{x_1, \dots, x_n\}$ y $\mathbf{Y} = \{y_1, \dots, y_m\}$ dos conjuntos de puntos en el plano. Escriba una función que calcule la función de distancia de Hausdorff $H(\mathbf{X}, \mathbf{Y}) = \max(h(\mathbf{X}, \mathbf{Y}), h(\mathbf{Y}, \mathbf{X}))$ en donde $h(\mathbf{X}, \mathbf{Y}) = \max_{x \in \mathbf{X}} \{\min_{y \in \mathbf{Y}} d(x, y)\}$ y $d(x, y)$ es la distancia Euclidiana entre x y y .

Ejemplo: ante la ejecución de la función con los siguientes conjuntos :

$$\mathbf{X} = \{(1, 1), (0, 0)\}, \mathbf{Y} = \{(1, -1), (0, 0)\},$$

se debe devolver $\sqrt{2}$.

Problema 5 [1.5 punto]

Escribir una función que determine si un numero entero pasado en parámetro es una potencia de 2 o no.

Ejemplo: ante la ejecución de la función con el entero 7 de entrada, se devolverá *false*.

Problema 6 [1.5 punto]

Suponga un sistema de ecuaciones como el siguiente:

$$\begin{cases} ax + by = c \\ dx + ey = f. \end{cases}$$

En el sistema previo a, b, c, d, e, f son constantes, mientras que x, y son las variables. Escriba una función **void solve(double a, double b, double c, double d, double e, double f, double &x, double &y, int &solutionType)**, que dados los valores de a, b, c, d, e y f , determine los valores de x e y que dan solución al sistema y los almacene en las variables correspondientes. En los casos en los que el sistema no tenga solución, almacene el valor 0 en solutionType; si el sistema tiene una única solución almacene el valor 1; finalmente si tiene un número infinito de soluciones almacene el valor 2 en solutionType.

Ejemplo: ante la ejecución de la función con los valores $a = 2, b = 1, c = 5, d = 1, e = 1, f = 3$, la función deberá almacenar el valor 2 en x , el valor 1 en y y el valor 1 en solutionType.

Problema 7 [2.0 punto]

Escriba una función **int countDifferentWays(int data[N], int S)** que dado un arreglo que contiene N números positivos **diferentes entre sí**, devuelva la cantidad de formas en que usando solo la operación sumar, se puede obtener el número S . Cada número se puede usar solo una vez. Si algún elemento del arreglo es igual a S , se debe contabilizar como una forma de conseguir el número S .

Ayuda: utilizar el hecho de que si consideras un dato d en el arreglo, entonces, se puede contar todas las maneras de obtener S argumentando que o una manera de sumar incluye a d , y eso lleva a contar el número de formas que se puede obtener $S - d$ con otros datos que d , o una manera de sumar no incluye a d , y eso lleva a contar el número de formas que se puede obtener S con otros datos que d .

Ejemplo: ante la ejecución de la función con $S = 7$ y el arreglo

$$data = \{1, 2, 5, 4, 9, 3, 6, 7\}$$

la función debe devolver 5. Las formas en que se puede conseguir son: $\{1 + 2 + 4, 1 + 6, 2 + 5, 4 + 3, 7\}$.