

GUÍA EXAMEN DE PROGRAMACIÓN

Instrucciones

El objetivo del Examen de Programación es saber si el sustentante cuenta con las Competencias Mínimas Necesarias para Desarrollar Software bajo el paradigma de programación estructurada y/o orientada a objetos que permita la resolución de problemas de Ingeniería a través del uso de computadoras.

Queremos resaltar que en el examen de programación, el énfasis no es tanto en la sintaxis del lenguaje utilizado, sino más bien en la estructura y la lógica interna del algoritmo empleado para resolver el problema. Los candidatos podrán usar el lenguaje de su elección para contestar a las preguntas; podrán también hacer uso de pseudo-código.

En algunos problemas, es posible que necesite ordenar un conjunto de datos. No es necesario que implemente un algoritmo de ordenación, sino que puede suponer que existe una función llamada sort, que recibe como parámetros un arreglo y el número de datos en el arreglo y los ordena. En lo referente a estructuras de datos, sólo tiene acceso a arreglos (de las dimensiones que quieran), cualquier otra cosa que quiera utilizar la debe implementar.

Contenido temático a Evaluar.

Unidad	Temas	Subtemas
1	Algoritmia	1.1 Definición. 1.2 Formulación y especificación 1.3 Técnicas de Especificación 1.3.1 Pseudocódigo 1.3.2 Diagramas de flujo 1.4 Pruebas 1.4.1 Pruebas de Escritorio 1.4.2 Verificación y validación formal
2	Elementos básicos del lenguaje	2.1. Estructura de un programa 2.2 Compilación y ejecución. 2.3. Variables 2.4 Operadores aritméticos, relacionales y lógicos 2.5 Estructuras de Control y Selección 2.6 Tipos de Datos
3	Estructuras de Datos	3.1 Arreglos. 3.1.1 Vectores 3.1.2 Matrices 3.2 Listas 3.2.1 Ligadas simples 3.2.2 Listas Ligadas Dobles

		3.3 Pilas 3.4 Colas 3.5 Árboles 3.6 Grafos
--	--	---

Bibliografía y Software de apoyo.

1. Kernighan, Brian W.; El lenguaje de programación C (2ª Ed); Ed. Prentice Hall; 1998
2. Stephen G. Kochan; Programming in C; 3rd; SAMS; 2004
3. Kernighan, Brian W.; The Practice of Programming; Ed. Addison-Wesley; 1999.
- 4 G. Levine G., Introducción a las computadoras y programación estructurada,. Ed. Mc Graw Hill
5. Peter Norton, Introducción a la computación, Ed. Mc. Graw Hill
6. Robert Sedgewick, Algorithms in C++, Ed. Addison Wesley
7. M. Morris Mano, Arquitectura de computadoras, Ed. Prentice Hall
8. O. J. Dahl, E. W. Dijatra, C.A.R. Huare, Programación estructurada, Ed.Tiempo contemporáneo
9. Byron S. Gottfried, Programación en C, Ed. Mc Graw Hill
10. Herbert Schildt, C manual de referencia, Ed. Mc Graw Hill
11. Jamsa, Lenguaje C bibliotecas de funciones, Ed. Mc Graw Hill
12. Programming Python. Mark Lutz. O'Reilly, 2010
13. Java, How to Program. Pau Deitel and Harvey Deitel Prentice Hall, 9th Edition, 2011
14. The Art of R Programming: A Tour of Statistical Software Design Norman Matloff

Ejemplo de Reactivos del Examen

A continuación se presentan algunos ejercicios tipo de muestra como vienen en el examen para que el sustentante.

Problema 1.

Realizar un programa que permita obtener el resultado la siguiente serie:

$$1^3+2^3+3^3+...+\square^3$$

Problema2.

Realizar un programa que permita realizar la suma de la siguiente serie: $1+3+5+...+(2n-1)$ si necesidad de utilizar ciclos.

Problema 3.

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 4.

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:

```
10 26
12 20
-5 4
```

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

Problema 5.

Escribir una función que tome de entrada un arreglo de enteros, y su tamaño, y que invierta el orden de los números contenidos en el arreglo, escribiendo el resultado en el mismo arreglo, y sin usar arreglo auxiliar. Por ejemplo, al pasar de entrada el arreglo: 15298

este mismo arreglo contendrá al salir de la función: 89251

Problema 6.

Dada una matriz A de enteros (representada como arreglo bi-dimensional), dar el código para guardar en un arreglo (uni-dimensional) aquellos elementos de A que sean negativos o nulos, y calcular el número de dichos elementos.

Problema 7.

¿Qué hace la siguiente función?

```
int mysterious(int N) {
    int i;
    int count = 0;
    for (i=0;i<=N;i++) {
        if (i%3==0){
            count+=i;
        }
    }
    return count; }
```