Prueba de programación

La prueba está constituida por preguntas de selección múltiple y ejercicios prácticos de programación; utilice el lenguaje de programación a su preferencia. No es necesario que utilice el mismo lenguaje para todas las preguntas.

El entregable de la prueba debe cumplir con los siguientes requisitos:

1. Se debe entregar un archivo comprimido que contenga todos los ejercicios realizados, este debe tener el nombre del participante.
2. Cada ejercicio deberá recibir unos parámetros de base y deberá responder por medio de un print en consola
3. **Para lenguajes de compilador**
   1. El código fuente debe venir en un archivo de texto con el nombre del ejercicio. *Ejemplo: fibonacci*
   2. En caso de utilizar librerías se debe incluir el código respectivo para su importe
   3. No se pueden utilizar librerías que no hagan parte del paquete base del lenguaje
   4. En caso de utilizar clases se deberán adicionar en el mismo archivo de texto con su denotación correspondiente
4. **Para lenguajes de interpretación**
   1. Se debe entregar el archivo en la extensión utilizada para cada lenguaje (.py, .vbs, etc.), este debe estar nombrado de acuerdo con el nombre del ejercicio. *Ejemplo: Fibonacci*
   2. No se pueden utilizar librerías que no hagan parte del paquete base del lenguaje
   3. El código necesario debe ser agrupado dentro de un mismo archivo
5. **Para** **consultas en SQL:** Complete bajo la pregunta utilizando este mismo documento en Word.
6. Para las preguntas de selección múltiple se deberá entregar este archivo en Word con la respuesta en su totalidad resaltada.

Preguntas

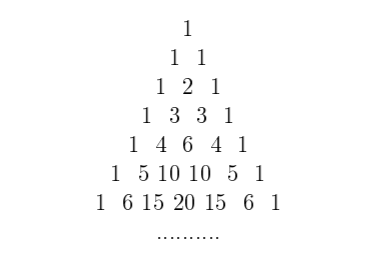
1. ¿Cuál de las siguientes estructuras de datos tiene una complejidad de búsqueda de O(1)?
   1. Tabla Hash
   2. Árbol binario
   3. Grafo
   4. Ninguna de las anteriores
   5. Todas las anteriores
2. De los siguientes algoritmos de ordenamiento, ¿Cuál es el más eficiente?
   1. Mergesort
   2. Bubblesort
   3. Insertion sort
   4. Quicksort
3. ¿Como se llama el diagrama que ayuda a representar la relación entre las tablas de una base de datos?
   1. Diagrama lógico
   2. Diagrama UML
   3. ETL
   4. Modelo relacional
4. **Fibonacci**

Cree una función que reciba un numero *n* e imprima un resultado acorde a los siguientes parámetros:

* 1. f(0) = 1
  2. f(1) = 1
  3. f(n) = f(n-1) + f(n-2)

1. **Triangulo** **de** **Pascal**

Para este ejercicio se utilizará el triángulo de pascal, el cual se define por niveles de la siguiente manera, el primer nivel cuenta con un 1, desde el segundo nivel se define por la suma de los números adyacentes en el nivel anterior y se completa con un 1 a cada uno de los extremos. Se puede ver un ejemplo en la siguiente imagen:



Cree una funcione que reciba dos parámetros f(*i,j*) y retorne el número ubicado en el nivel *i* y en la posición de izquierda a derecha *j*

1. **Máximo**

Cree una función que reciba como parámetros un arreglo de números y dos enteros (*a,b*) , la función debe evaluar el número máximo del arreglo entre las posiciones *a* y *b*

1. **Ordenamiento**

Cree una función que reciba un arreglo de palabras en un orden indeterminado, ordene las letras de cada una de las palabras en orden alfabético y retorne un arreglo con las palabras reorganizadas.

1. **Escalera N**

Se cuenta con una escalera de N escalones, en cada paso se puede subir ya sea de a 1 escalón o de a 2 escalones. Dado el valor N como parámetro cree una funciona que calcule la cantidad de maneras distintas en las cuales puede subir esta escalera.

*El orden importa*

Como ejemplo, si N = 4 se tienen 5 maneras distintas de subir la escalera