

OnServices

making safe & easy your business

PRUEBAS SELECCIÓN

Empresa Inscrita en catalogo INCIBE

(Instituto Nacional de Ciberseguridad del gobierno de España desde 2009)

Pruebas Selección

TÉCNICO JUNIOR

Contacto: 91 141 96 57

RESUMEN

DESCRIPCIÓN Documento	Pruebas de selección
NOMBRE Y APELLIDOS CANDIDAT@	Lighuen Lacamoire
FECHA	2 de agosto 2023
INSTRUCCIONES	<p>Es necesario que el candidat@ cree un repositorio online en plataforma gratuita github con los ejercicios requeridos y que sea un repositorio PRIVADO donde se dará acceso como colaborador a nuestra cuenta de revisión: selecciongithub@onservices.es incluyendo los permisos a reportes, seguimiento de errores y gestión de proyectos en dicho repositorio de pruebas.</p> <p>La estructura de los directorios en el repositorio de pruebas y la localización de los ficheros deberá mantener la siguiente estructura básica:</p> <p>Nombredelcandidat@-numerodeejercicio-onservices/</p> <p>Ejemplo para el ejercicio 1 y nombre ficticio de candidat@ Pepe Grillo</p> <p>pepegrillo-1-onservices/</p> <p>Importante: El acceso a nuestra cuenta de revisión SOLO debe darse una vez el candidat@ haya terminado TODOS los ejercicios.</p> <p>En dicho repositorio de pruebas el candidat@ puede crear ficheros TXT donde exponga observaciones que considere necesarias en cada ejercicio.</p> <p>*Será necesario que el candit@ responda a tod@s los ejercicios / cuestiones planteadas para dar por válida las pruebas.</p> <p>SE VALORARÁ POSITIVAMENTE QUE SE RESPONDA A TOD@S LOS EJERCICIOS, AUNQUE EL RESULTADO NO SEA EL CORRECTO.</p> <p>LOS EJERCICIOS DE PROGRAMACION SE VAN A EJECUTAR EN UN IDE COMO ECLIPSE, POR LO CUAL RECOMENDAMOS DESARROLLAR EN ECLIPSE O VISUAL EN EL CASO DE C#. POR FAVOR ADJUNTAR TODOS LOS PAQUETES.</p>

EJERCICIOS

1. Crea una aplicación en que solicite un string por pantalla y compruebe si se trata de un palíndromo o no ASÍ como la salida entre DOS tipos de codificación distintos a elección del candidat@. (Lenguajes de programación admitidos para este ejercicio Java - C#)

Ejemplo:

Salida 1:

La salida para el string "Urú" sería: Es palíndromo

La salida para el string "coche" sería: No es palíndromo

Salida 2:

La salida para el string "Urú" sería: Urú – UTF8

La salida para el string "Urú" sería: Urú – ISO-8859-1

2. Crear una aplicación/solución para encontrar todos los números Armstrong de tres dígitos. Los números Armstrong son los números de tres dígitos para los que la suma de sus tres dígitos elevados al cubo es igual al número de tres dígitos. Por ejemplo, el número 153 es un número Armstrong porque $1^3 + 5^3 + 3^3 = 153$. Sin embargo, 294 no lo es porque $2^3 + 9^3 + 4^3 = 801$ (no 294). @. (Lenguajes de programación admitidos para este ejercicio Java - C# - C++ - JavaScript)

La salida del programa debería ser inicialmente:

153

370

371

407

3. Crea un programa/solución que tenga una clase padre *Figura* con los métodos *double area()* y *double perimetro()*. También tendrá que haber una clase *Círculo*, una clase *Rectángulo* y una clase *Cuadrado*. El programa deberá preguntar al usuario por pantalla el tipo de figura para la que quiere conocer el área y el perímetro, solicitándole los datos necesarios por pantalla. (Lenguajes de programación admitidos para este ejercicio Java)

Ejemplos:

Menú

1. Círculo
2. Rectángulo
3. Cuadrado
Indica tu selección: 1
Indica el radio: 3.5
El área del círculo es: 38.465
El perímetro del círculo es: 21.98

Menú

1. Círculo
2. Rectángulo
3. Cuadrado
Indica tu selección: 2
Indica la longitud: 4
Indica el ancho: 2
El área del rectángulo es: 8
El perímetro del círculo es: 12

Menú

1. Círculo
2. Rectángulo
3. Cuadrado
Indica tu selección: 4
Indica el lado: 2.2
El área del cuadrado es: 4.840000000000001
El perímetro del cuadrado es: 8.8

4. Implementa una solución al siguiente problema creando una aplicación: (Lenguajes de programación admitidos: A elección del candidat@ de todos los existentes en el mercado profesional)

Se necesita implementar un algoritmo para que, dado un conjunto de pares de valores, que representan el peso y la altura de un grupo de personas, se obtenga un listado con el mayor número de personas posible en el que se cumpla que cada individuo tiene mayor peso y altura que el inmediatamente anterior en el listado.

Los datos de entrada serán leídos de un fichero de texto plano externo, en el que cada línea representa la altura en cm y el peso en kg de una persona, separados por comas y sin espacios. No es necesario implementar un control de errores del fichero, se asume que siempre es proporcionado en el formato correcto.

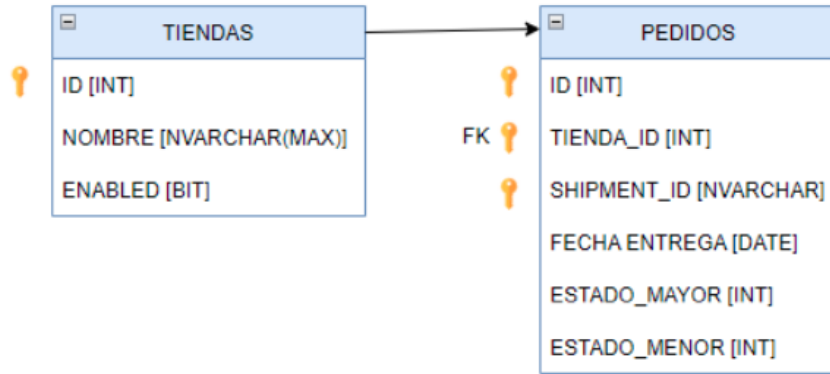
El listado resultante se sacará por pantalla.

Ejemplo:

78,189
75,178
80,185
90,170
92,192

Solución: En pantalla se mostrará el listado con las líneas 2-3-5

5. Teniendo la siguiente definición de tablas, cual sería la query necesaria en un motor de base de datos transaccional para obtener por cada tienda tanto la fecha de fecha entrega como el número total de pedidos ordenados por día y nombre de tienda alfabéticamente.



ID	NOMBRE	ENABLED
1	Ciudad Imagen	Y
2	Alcobendas	Y
3	El Pinar	Y
4	Cuenca	Y

ID	TIENDA_ID	SHIPMENT_ID	FECHA ENTREGA	ESTADO_MAYOR	ESTADO_MENOR
1	4	12345678	2022-05-13	700	650
2	2	87654321	2022-05-16	600	650
3	2	01593657	2022-05-17	650	650
4	1	15935746	2022-05-15	300	600

Ejemplo de resultado a obtener con datos dummy:

TIENDA	FECHA ENTREGA	Nº PEDIDOS
Alcobendas	2022-05-13	2
Ciudad Imagen	2022-05-13	81
El Pinar	2022-05-13	8
Ciudad Imagen	2022-05-14	36
El Pinar	2022-05-14	3
Ciudad Imagen	2022-05-16	9
El Pinar	2022-05-16	2
Alcobendas	2022-05-17	1
Ciudad Imagen	2022-05-17	13
El Pinar	2022-05-17	2
Ciudad Imagen	2022-05-18	3

CONFIDENCIAL

