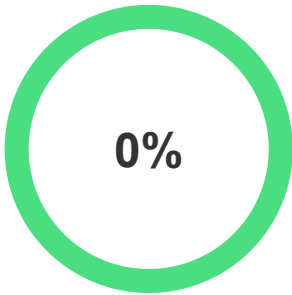


Team Eureka 3it

Romina

Fecha Inicio: 3/9/2025 11:46:50 Fecha Fin: 3/9/2025 11:57:44



Tiempo Total: 0 Horas 10 Minutos 54 Segundos

Percentil: 0%

| Información Candidato | Información Prueba | Información Instancia |
|----------------------------------|--|-----------------------------------|
| Id: 24 | Nombre/ID: Team Eureka 3it | Tiempo Límite: 120 minutos |
| Nombre: Romina | Descripción: Prueba técnica diseñada para los postulantes al Team Eureka. | Tipo: TST |
| Email: ro.saezp@duocuc.cl | | |

Resumen por Categoría

| Descripción | Resultado | Puntaje (Valor) |
|---|---------------|---------------------|
| Programación Resultado Categoría: | | 0/60 |
| | Puntaje > 66% | 33% < Puntaje < 66% |
| | | Puntaje < 33% |

Resumen por Pregunta

| Descripción | Categoría | Nivel de Dificultad | Puntaje |
|---|--------------|---------------------|---------|
| 1.- Encuentra el mayor número en un arreglo | Programación | Fácil | 0 de 10 |
| Descripción: Tienes un arreglo (llamado myArray) con 5 elementos (enteros en el rango de 1 a 100). Escribe un programa utilizando TLang | | | |

que imprima el número más alto del arreglo (Si se repite, solo imprimir una vez). El código que llena el arreglo ya está escrito, pero puedes editarlo para probar con otros valores. Con el botón de refrescar puedes recuperar el valor original que será utilizado para evaluar la pregunta como correcta o incorrecta durante la ejecución. **IMPORTANTE:** Asegúrate de solo imprimir el número y ningún otro texto, ya que esto marcaría la pregunta como inválida. Por ejemplo, para el arreglo indicado, el output del programa debe ser: 35

Respuesta del candidato:

```
maxValue := myArray[1]
for i:=2 to 5
    if myArray[i] > maxValue then
    endif
next
PrintIn maxValue
```

2.- Encuentra el número que más se repite en un arreglo

Programación

Medio

0 de 20

Descripción:

Tienes un arreglo (llamado myArray) con 10 elementos (enteros en el rango de 1 a 5). Escribe un programa utilizando TLang que imprima el número que más se repite del arreglo y también imprimir la cantidad de veces que aparece. El código que llena el arreglo ya está escrito, pero puedes editarlo para probar con otros valores. Con el botón de refrescar puedes recuperar el valor original que será utilizado para evaluar la pregunta como correcta o incorrecta durante la ejecución. Su programa debe analizar el arreglo de izquierda a derecha para que en caso de que dos números tengan el mismo número de ocurrencias, el que aparece por primera vez de izquierda a derecha será el que se imprima. La salida de los datos para el ejemplo sería la siguiente: Rep: 3 Val: 5 En el ejemplo, hay 3 ocurrencias del número 5 en el arreglo. **IMPORTANTE:** Asegúrate de que el output del programa sea exactamente como se muestra en el ejemplo, ya que cualquier diferencia hará que la respuesta se marque como incorrecta.

Respuesta del candidato:

```
maxCount := 0
mostFrequent := myArray[1]

for i =: 1 to 10
    count := 0
    for j :=
```

3.- Ordenar bloques de números

Programación

Difícil

0 de 10

Descripción:

Tienes un arreglo de números enteros desordenados. Escribe un programa utilizando TLang que ordene el arreglo de menor a mayor e imprima el resultado. El código que llena el arreglo ya está escrito, pero puedes editarlo para probar con otros valores. Con el botón de refrescar puedes recuperar el valor original que será utilizado para evaluar la pregunta como correcta o incorrecta durante la ejecución. El programa debe imprimir los números ordenados separados por espacios, en una sola línea. Por ejemplo, para el arreglo [64, 34, 25, 12, 22], el output del programa debe ser: 12 22 25 34 64

IMPORTANTE: Asegúrate de imprimir los números separados por espacios exactamente como se muestra en el ejemplo, ya que cualquier diferencia hará que la respuesta se marque como incorrecta.

Respuesta del candidato:

dxgvvxx

4.- Consulta SQL con rango de números

Programación

Fácil

0 de 10

Descripción:

Escribe una consulta SQL que seleccione todos los registros de la tabla 'empleados' donde el campo 'salario' esté entre 3000 y 5000 (inclusive). La tabla empleados tiene la siguiente estructura: - id (INTEGER) - nombre (VARCHAR) - departamento (VARCHAR) - salario (INTEGER) Tu consulta debe retornar todos los campos de los empleados que cumplan con el criterio de salario. Ejemplo de resultado esperado: Empleados con salario entre 3000 y 5000. IMPORTANTE: Escribe solo la consulta SQL, sin explicaciones adicionales.

Respuesta del candidato:

```
SELECT firstname , lastname, salary
FROM APPX_employee
WHERE salary >= 500000 AND salary <= 100000
ORDER BY firstname ASC;
```

5.- Consulta SQL con agrupación y filtrado

Programación

Medio

0 de 10

Descripción:

Escribe una consulta SQL que agrupe los empleados por departamento, cuente cuántos empleados hay en cada departamento y muestre solo aquellos departamentos que tienen más de 2 empleados. Los resultados deben estar ordenados por cantidad de empleados de forma descendente. La consulta debe mostrar: - Nombre del departamento - Cantidad de empleados (con alias 'cantidad') IMPORTANTE: Escribe solo la consulta SQL, sin explicaciones adicionales.

Respuesta del candidato:

```
SELECT d.departament_name, COUNT(e.id) AS employee_count
FROM APP_employee e
JOIN APPX_department d ON e.department_id = d.id
GROUP BY d.department_name
HAVING COUNT(e.id) >= 2
ORDER BY d.department_name ASC;
```

No Respondido

Parcialmente Correcto

Incorrecto

Correcto