



Team Eureka 3IT

Guillermo Diaz Quezada

Fecha Inicio: 2024-09-10 16:33:40 Fecha Fin: 2024-09-10 18:01:03

Puntaje Total:

Resultado:

59/60

Tiempo Total: 1 Hora 27 Minutos 23 Segundos

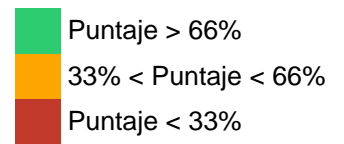
Percentil: 100%



Información Candidato	Información Prueba	Información Instancia
<p>Id: 484386</p> <p>Nombre: Guillermo Diaz Quezada</p>	<p>Nombre/ID: Team Eureka 3IT / 3324</p> <p>Descripción Prueba técnica diseñada para los postulantes al Team Eureka</p>	<p>Límite de Días: 2</p> <p>Tiempo Límite: 120 minutes</p> <p>Tipo: TST</p> <p>Proceso: Eureka 3IT</p>

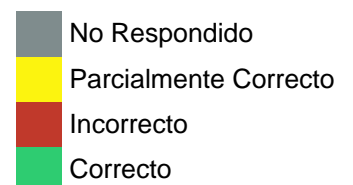
Resumen por Categoría

Descripción	Resultado	Puntaje (Valor)
Programación		39/40
Resultado Categoría:		
SQL		20/20
Resultado Categoría:		



Resumen por Pregunta

Descripción	Categoría	Nivel Dificultad	de Puntaje
1.- Encuentra el mayor número en un arreglo	PROG	Fácil	10 de 10
Escenarios de Validación:	<div><div>100 %</div></div>		
2.- Encuentra el número que mas se repite en un arreglo	PROG	Medio	19 de 20
Escenarios de Validación:	<div><div>100 %</div></div>		
Eficiencia del Código:	<div><div>90 %</div></div>		
3.- Ordenar bloques de números en arreglo	PROG	Difícil	10 de 10
Escenarios de Validación:	<div><div>100 %</div></div>		
4.- Pregunta SQL where simple rango de números	SQL	Fácil	10 de 10
Escenarios de Validación:	<div><div>100 %</div></div>		
5.- Pregunta SQL con order by, group by y having	SQL	Medio	10 de 10
Escenarios de Validación:	<div><div>100 %</div></div>		



Detalles por Pregunta

1.- Encuentra el mayor número en un arreglo

Tipo:	Categoría	Puntaje
Pregunta de programación con código predefinido	PROG	10 de 10
Nivel de Dificultad	Familia:	SubFamilia:
Fácil	Programación	Programación General

Texto de la Pregunta

Tienes un arreglo (llamado myArray) con 5 elementos (enteros en el rango de 1 a 100). Escribe un programa utilizando TLang (para ver la sintaxis haga clic en el botón con la i en la esquina superior derecha) 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:

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

Println maxValue
```

Escenarios de Validación

Descripción	Estado
Caso Básico	Correcto
Numero Doble	Correcto
Respuesta sin considerar Formato	Correcto

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

Tipo:
Pregunta de programación con

codigo predefinido

Nivel de Dificultad

Medio

Familia:

Programación

SubFamilia:

Programación General

Texto de la Pregunta

Tienes un arreglo (llamado myArray) con 10 elementos (enteros en el rango de 1 a 5). Escribe un programa utilizando TLang (para ver la sintaxis haga clic en el botón con la i en la esquina superior derecha) 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. Tenga en cuenta que el código que escriba debe imprimir los resultados exactamente como se muestra con el fin de que la pregunta sea considerada válida.

El código para declarar y poblar myArray ya está ahí, puede editarlo para probar con otros valores y puede hacer clic en el botón de actualizar junto a él para volver al valor original que se utilizará para validar su código durante la prueba.

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:

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

for i := 1 to 10
  count := 0
  for j := 1 to 10
    if myArray[i] = myArray[j] then
      count := count + 1
    endif
  next
  if count > maxCount then
    maxCount := count
    mostFrequent := myArray[i]
  endif
next

Println 'Rep: ' + maxCount
Println 'Val: ' + mostFrequent
```

Escenarios de Validación

Descripción	Estado
Numero Doble	Correcto
Caso Básico	Correcto
Caso Probar Orden Secuencia	Correcto
Caso Respuesta Aproximada	Correcto

3.- Ordenar bloques de números en arreglo

Tipo:	Categoría	Puntaje
Pregunta de programación con código predefinido	PROG	10 de 10

Nivel de Dificultad	Familia:	SubFamilia:
Difícil	Programación	Programación General

Texto de la Pregunta

Se tiene un arreglo myArray que contiene bloques de números. Los bloques pueden ser de cualquier largo, los números contenidos están en el rango de 1 a 9 y se separan por un cero para definir los bloques. Se deben ordenar los números de los bloques individualmente de menor a mayor e imprimir las secuencias separando los bloques por un espacio. Por ejemplo, para el arreglo: (1,3,2,0,7,8,1,3,0,6,7,1) la respuesta esperada es:

123 1378 167

El arreglo y su longitud están definidos en la sección de código predefinido. Asumir que este código predefinido puede variar (valores y longitud del arreglo) y se debe tener en cuenta lo siguiente al programar:

- El número de bloques es variable.
- Un cero marca el final de un bloque y el inicio de otro. El inicio del arreglo se asume como el inicio de un bloque y el final de un arreglo se asume como el final de un bloque (Por lo tanto un cero al inicio o al final del arreglo de hecho implicarían un bloque sin elementos)
- Un bloque puede no contener elementos, en cuyo caso se imprimirá una x. Por ejemplo, para (2,1,0,0,3,4) se imprimiría.

12 X 34

Para escribir el programa debes usar TLang, un lenguaje con solo un par de instrucciones. Para la sintaxis hacer clic en la i en la parte superior derecha. El puntaje de la pregunta se determina por obtener resultados correctos en diversos escenarios y por la eficiencia del código (cuanto menos instrucciones ejecute el programa, se considera más eficiente. Por ejemplo, un doble loop de 10 iteraciones generaría 100 ejecuciones). No es necesario validar ninguna condición de borde no mencionada aquí (por ejemplo, no es necesario validar que n sea mayor que cero)

Respuesta:

```
array tempArray
tempLength := 0
isEmptyBlock := true

for i := 1 to n
  if myArray[i] = 0 then
    if isEmptyBlock = true then
      Print 'X '
    else
      for j := 1 to tempLength
        for k := 1 to tempLength - 1
          if tempArray[k] > tempArray[k + 1] then
            temp := tempArray[k]
            tempArray[k] := tempArray[k + 1]
            tempArray[k + 1] := temp
          endif
        next
      next
      for j := 1 to tempLength
        Print tempArray[j]
      next
      Print ' '
    endif
    tempLength := 0
    isEmptyBlock := true
  else
    tempLength := tempLength + 1
    tempArray[tempLength] := myArray[i]
    isEmptyBlock := false
  endif
next

if isEmptyBlock = true then
  Print 'X'
else
  for j := 1 to tempLength
    for k := 1 to tempLength - 1
      if tempArray[k] > tempArray[k + 1] then
        temp := tempArray[k]
        tempArray[k] := tempArray[k + 1]
        tempArray[k + 1] := temp
      endif
    next
  next
  for j := 1 to tempLength
    Print tempArray[j]
```

```
next
endif
```

Escenarios de Validación

Descripción	Estado
Caso Básico	Correcto
Caso Grande	Correcto
Caso Borde Inicio	Correcto
Caso Borde Fin	Correcto
Caso Borde Tamaño Arreglo y vacios	Correcto
Caso Vacío al Medio	Correcto

4.- Pregunta SQL where simple rango de números

Tipo:	Categoría	Puntaje
Pregunta de programación sin código predefinido	SQL	10 de 10

Nivel de Dificultad	Familia:	SubFamilia:
Fácil	Programación	SQL y Bases de Datos

Texto de la Pregunta

Obtener utilizando una consulta SQL el nombre, apellido y salario (firstname, lastname y salary) para los empleados con un salario mayor o igual a 50000 y menor o igual a 100000. Mostrar los resultados ordenados por nombre de forma ascendente.

Tablas:

APPX_employee (id, firstname, lastname, department_id, salary, educationlevel_id)

APPX_department (id, department_name, department_city)

APPX_educationlevel (id, description)

Respuesta:

```
SELECT firstname, lastname, salary
FROM APPX_employee
WHERE salary >= 50000 AND salary <= 100000
ORDER BY firstname ASC;
```


Escenarios de Validación

Descripción	Estado
Simple where range	Correcto

5.- Pregunta SQL con order by, group by y having

Tipo:	Categoría	Puntaje
Pregunta de programación sin código predefinido	SQL	10 de 10
Nivel de Dificultad	Familia:	SubFamilia:
Medio	Programación	SQL y Bases de Datos

Texto de la Pregunta

Obtener utilizando una consulta SQL el listado de departamentos que tienen 2 o más empleados. Indicar el nombre del departamento (campo department_name) y la cantidad de empleados, ordenado por el nombre del departamento de forma ascendente.

Tablas:

APPX_employee (id, firstname, lastname, department_id, salary, educationlevel_id)

APPX_department (id, department_name, department_city)

APPX_educationlevel (id, description)

Respuesta:

```
SELECT d.department_name, COUNT(e.id) AS employee_count
FROM APPX_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
```

Escenarios de Validación

Descripción	Estado
Group by y Having	Correcto