

Introducción a la Programación Ejercicios

1. Responda las siguientes preguntas:

A. **(4 puntos)** Para que sirve la precedencia de operadores en Java?

B. **(4 puntos)** Cual es la ventaja de que un arreglo reserve espacio continuo en memoria?

C. **(4 puntos)** A que nos referimos cuando hablamos de “type-checking” en lenguajes de programación?

D. **(4 puntos)** Agregue paréntesis a las siguientes expresiones de forma que al evaluarlas den el resultado de la derecha.

Expresión: $4 / 3 / 2 - 3$
Expresión: $-10 * -4 + 2$

Resultado Esperado: -2
Resultado Esperado: 60

E. **(4 puntos)** Considerando que $a=1$; $b=2$ y $n=1$. Indique cual es el resultado de evaluar las siguientes expresiones.

Expresiones

Resultado

$1 \leq 1$

$\text{true} == (1 < 1)$

$(a + b) < 3 * a$

$(n > 2) || (n == 6)$

2. (10 puntos) Escriba un programa que lea 3 enteros positivos, que representan el tamaño de 3 barras. El programa debe determinar si una persona puede formar un triángulo utilizando esas tres barras. **Pista:** Se sabe que 3 números forman un triángulo si son positivos y la suma de 2 cualesquiera de ellos es mayor que el tercero. El programa debe imprimir en consola "es posible formar un triángulo" o "no es posible formar un triángulo según corresponda".

3. (10 puntos) Defina una función `div2`, que permita verificar si un numero es divisible entre 2. Utilizando la siguiente propiedad: se dice que un numero es divisible entre 2 si el ultimo de sus dígitos es par. **Por ejemplo:** `div2(78) -> true`, porque 8 es par.

4. (10 puntos) Escriba un programa que cuente el numero de palabras de una cadena ingresada por consola. Puede asumir que las palabras dentro de la cadena están separadas exactamente por un espacio. El programa debe imprimir el número de palabras en consola.

5. (20 puntos) Defina una función que permita determinar si un numero es perfecto o no. Un entero positivo es perfecto si la suma de sus divisores es igual al mismo numero. **Por ejemplo:** 6 es perfecto porque $1+2+3=6$. Al respecto, escriba una función que reciba un número entero positivo y entregue true (verdadero) si es perfecto o false si no lo es.

```
static boolean perfecto(int n) {
```

```
}
```

6. (15 puntos) Escriba una función que verifique si los elementos de un arreglo están ordenado ascendentemente.

```
static boolean estaOrdenado(int[] array) {
```

```
}
```

7. (15 puntos) Hacer un programa que lea "n" números de consola e imprima los 3 números mas grandes.