**Practica 2**

**Martin Valencia Vallejo y Daniel Montoya Arenas**

Delivery of the solution is done in two ways:

I. Through the site set up in the OnlineGDB Classroom by clicking on this link. Each member of the team must enter the link and deliver the proposed solution.

II. Include a text document with the names of the two team members and the answers to the following points:

1. Describe in detail the process you carried out to solve the exercise.

2. Write a form in which the currying process is manifested in the proposed solution.

3. For each function developed, indicate whether you identify it as polymorphic, of higher order, or, if you do not consider it to be classified in one or both criteria, justifying each defined determination.

4. Store the developed source file(s) in a GitHub repository, invite the teacher as a collaborator and place a link to access the developed repository at this point in the written document.

5. For the presentation of the developed solution, the team must elaborate a video in which each member appears detailing step by step the process that was carried out, all team members must appear in the video and participate. The video must be uploaded to the Internet and a link must be generated so that it can be accessed for the corresponding review of the delivery. The duration of the video should not exceed seven (7) minutes. This link must be included in this point of the document.

**Solución:**

**1.**

**Estructura General del Código:**

El código consta de varias funciones definidas, cada una con un propósito específico en el proceso de clasificación de números de admisión. Estas funciones incluyen operaciones para tomar partes de un número, buscar divisores, identificar categorías basadas en los divisores, y otras operaciones de manipulación de números.

**Funciones de Manipulación de Números:**

Las primeras funciones definidas son tomarTresUlt, tomarTresPrim, y tomarDos, las cuales se utilizan para extraer partes específicas de un número. tomarTresUlt toma los últimos tres dígitos de un número decimal, tomarTresPrim toma los primeros tres dígitos, y tomarDos toma los dos dígitos en la posición central.

**Funciones de Búsqueda de Divisores:**

La función buscarDivisores encuentra todos los divisores de un número dado. Utiliza una función auxiliar divisoresRec que realiza la búsqueda de divisores de manera recursiva. Estas funciones son esenciales para la identificación de ciertas características de los números de admisión.

**Identificación Basada en Divisores:**

La función identificar2 utiliza los divisores de un número para clasificarlo en una de las tres categorías: Ingeniería, Humanidades o Administrativo. Esta clasificación se realiza comparando la suma de todos los divisores excepto el último con el último divisor.

**Funciones de Conversión de Tipo:**

Se proporcionan funciones para convertir números entre tipos Double e Int. Estas funciones son importantes para manipular los números de admisión, ya que algunas operaciones requieren tipos específicos.

**Funciones para Completar Números de Admisión:**

Las funciones completarProgramaAcademico y completarNumeroAdmisiones completan los números de admisión con dígitos faltantes según ciertas reglas. Estas reglas incluyen la adición de dígitos para representar el programa académico y el número de admisiones.

**Funciones de Identificación y Verificación:**

La función identificar1 utiliza las funciones anteriores para identificar el tipo de programa académico basado en los divisores de un número de admisión. Por otro lado, verificarPar determina si un número es par o impar y devuelve el resultado como una cadena.

**2.**

La currificacion la podemos encontrar en la función **completarNumeroAdmisiones1**. Esta función toma un solo argumento, y, y produce un nuevo valor basado en ese argumento. Sin embargo, en lugar de tomar todos los argumentos necesarios para calcular el resultado final simultáneamente, esta función toma solo uno de los argumentos y devuelve otra función que espera el resto de los argumentos. Este enfoque permite que la función sea parcialmente aplicada, lo que significa que puede ser llamada con menos argumentos de los que espera originalmente.

**Función Principal:**

La función main es la función principal del programa. Solicita al usuario un número de admisión, luego completa y verifica el número según las reglas definidas. Luego, utiliza varias funciones para extraer información del número de admisión y generar una cadena de salida que describe el tipo de programa académico y si el número es par o impar.

**3.**

**tomarTresUlt:**

Polimórfica: No es polimórfica. Esta función toma un argumento de tipo Double y devuelve un resultado de tipo Int. No hay parámetros genéricos ni uso de tipos polimórficos.

Orden superior: Tampoco es de orden superior. No toma funciones como argumentos ni devuelve funciones.

**tomarTresPrim:**

Polimórfica: Al igual que la función anterior, no es polimórfica. Toma un Double y devuelve un Int.

Orden superior: No es de orden superior. No involucra funciones de alto nivel.

**tomarDos:**

Polimórfica: No es polimórfica. Toma un argumento de tipo Int y devuelve un resultado de tipo Int.

Orden superior: No es de orden superior. No utiliza funciones de alto nivel.

**buscarDivisores:**

Polimórfica: No es polimórfica. Toma un Int y devuelve una lista de Int.

Orden superior: No es de orden superior. No utiliza funciones de alto nivel.

**divisoresRec**:

Polimórfica: Tampoco es polimórfica. Toma tres argumentos de tipo Int y devuelve una lista de Int.

Orden superior: En esta función es en donde más potencial hay para usar funciones de orden superior, por ejemplo una función map

**identificar2:**

Polimórfica: No es polimórfica. Toma una lista de Int y devuelve un String.

Orden superior: No es de orden superior. No utiliza funciones de alto nivel.

**doubleToInt:**

Polimórfica: No es polimórfica. Toma un Double y devuelve un Int.

Orden superior: No es de orden superior. No involucra funciones de alto nivel.

**enteroADouble:**

Polimórfica: No es polimórfica. Toma un Int y devuelve un Double.

Orden superior: No es de orden superior. No utiliza funciones de alto nivel.

**completarProgramaAcademico:**

Polimórfica: No es polimórfica. Toma un Double y devuelve un Int.

Orden superior: No es de orden superior. No involucra funciones de alto nivel.

**verificarProgramaAcademico:**

Polimórfica: No es polimórfica. Toma un Double y devuelve un Int.

Orden superior: No es de orden superior. No utiliza funciones de alto nivel.

**completarNumeroAdmisiones1:**

Polimórfica: No es polimórfica. Toma un Int y devuelve un Int.

Orden superior: No es de orden superior. No involucra funciones de alto nivel.

**completarNumeroAdmisiones2:**

Polimórfica: No es polimórfica. Toma un Int y devuelve un Int.

Orden superior: No es de orden superior. No utiliza funciones de alto nivel.

**completarNumeroAdmisiones:**

Polimórfica: Esta es polimórfica parcialmente ya que ella acepta cualquier valor, pero a las funciones internas no

Orden superior: No es de orden superior. No involucra funciones de alto nivel.

**identificar1:**

Polimórfica: No es polimórfica. Toma un Int y devuelve un String.

Orden superior: No es de orden superior. No utiliza funciones de alto nivel.

**verificarPar:**

Polimórfica: No es polimórfica. Toma un Int y devuelve un String.

Orden superior: No es de orden superior. No involucra funciones de alto nivel.

**4. Link al repositorio:**

[Daniel-montoya283/Practica2\_Lenguales\_Programacion (github.com)](https://github.com/Daniel-montoya283/Practica2_Lenguales_Programacion)