

Tarea Extraclase de Introducción a la Programación

Orientaciones generales:

El objetivo es que usted como estudiante sea capaz de desarrollar una aplicación de consola utilizando el lenguaje de programación Java que resuelva la situación planteada en el ejercicio que usted seleccione para desarrollar. Para realizar el ejercicio usted no solo debe demostrar conocimiento de programación sino ser capaz de combinarlo con conocimiento de Matemática I, Matemática computacional y por supuesto demostrar creatividad e ingenio.

La solución (la aplicación) debe entregarse (el código fuente, en la plataforma EVA en el espacio de la asignatura o curso; se habilitará un recurso para su envío) y discutirse (esto siempre de ser posible) dos semana antes del examen final (fecha aún por definirse).

Para solución puede utilizar cualquier entorno integrado de desarrollo que le permita y sea de su agrado para desarrollar en Java. En cuanto la diversidad de recursos que ofrece Java como lenguaje de programación se debe utilizar los siguientes:

- Variables y tipos de datos nativos
- Estructuras alternativas o bifurcaciones
- Estructura repetitivas o cíclicas
- Arreglos unidimensionales, bidimensionales (matrices) o de una dimensión superior
- Funciones cualquiera de sus variantes incluyendo funciones recursiva.

Cualquier otro elemento fuera de los anteriormente mencionados el estudiante puede utilizarlo pero debe demostrar dominio del mismo en la series de preguntas que se le harán.

Cada estudiante de seleccionar un ejercicio y no puede existir dos estudiantes realizando el mismo ejercicio.

A continuación en la próxima página cada uno de los ejercicios.

Ejercicio 1. La regla de Cramer es un teorema del álgebra lineal que da la solución de un sistema lineal de ecuaciones en términos de determinantes. Recibe este nombre en honor a Gabriel Cramer (1704 - 1752). Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java que sea capaz de resolver sistemas lineal de ecuaciones de hasta 4 variables utilizando esta regla. La aplicación debe ser capaz:

- Que el usuario escoja cuantas variables tiene el sistema lineal que desea resolver, introducir los coeficientes de las ecuaciones que componen el sistema.
- Imprimir el sistema de ecuaciones lineales con que se esta operando.
- Cambiar alguno de los coeficientes del sistema de ecuaciones lineales.
- Calcular los valores de las variables siempre y cuando el sistema tenga solución en caso de no tener se debe informar al usuario a través de un mensaje.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 2. En matemáticas, la eliminación de Gauss-Jordan, llamada así debido a Carl Friedrich Gauss y Wilhelm Jordan, es un algoritmo del álgebra lineal para determinar las soluciones de un sistema de ecuaciones lineales, encontrar matrices e inversas. Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java que sea capaz de resolver sistemas lineal de ecuaciones de hasta 4 variables utilizando esta regla. La aplicación debe ser capaz:

- Que el usuario escoja cuantas variables tiene el sistema lineal que desea resolver, introducir los coeficientes de las ecuaciones que componen el sistema.
- Imprimir el sistema de ecuaciones lineales con que se esta operando.
- Cambiar alguno de los coeficientes del sistema de ecuaciones lineales.
- Calcular los valores de las variables siempre y cuando el sistema tenga solución en caso de no tener se debe informar al usuario a través de un mensaje.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 3. Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java el cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre vectores con valores numéricos :

- Suma de dos vectores. Ambos vectores deben ser proporcionados por el usuario. La aplicación debe validar que dicha operación se puede efectuar de ser posible le debe mostrar el resultado al usuario de lo contrario mostrar un cartel con la explicación de porque no se pudo efectuar la operación.
- Producto vectorial de dos vectores. Ambos vectores deben ser proporcionados por el usuario. La aplicación debe validar que dicha operación se puede efectuar de ser posible le debe mostrar el resultado al usuario de lo contrario mostrar un cartel con la explicación de porque no se pudo efectuar la operación.
- Multiplicación de un vector por un escalar. Tanto el vector como el escalar debe ser proporcionado por el usuario.
- Ángulo comprendido entre dos vectores. Ambos vectores deben ser proporcionados por el usuario. La aplicación debe validar que dicha operación se puede efectuar de ser posible le debe mostrar el resultado al usuario de lo contrario mostrar un cartel con la explicación de porque no se pudo efectuar la operación. El resultado se debe mostrar en radianes y grados sexagesimales.
- Dados dos vectores proporcionados por el usuario determinar si estos son perpendiculares entre si.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 4. Se tiene una ecuación polinomial de la forma $ax^n+bx^{n-1}+...+yx+z=0$ y se quiere realizar una aplicación en consola utilizando el lenguaje de programación de Java la cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre la ecuación polinomial:

- Definir el grado de la ecuación y cada uno de los coeficientes de la ecuación polinomial. Una vez que el usuario termine esta opción se le debe mostrar la ecuación para que compruebe si es la ecuación con la que desea trabajar.
- Determinar si la ecuación polinomial tiene algún cero dentro de un rango. El rango debe ser definido por el usuario con un limite inferior y un limite superior.
- Cambiar alguno de los coeficiente de la ecuación. Una vez que el usuario termine esta opción se le debe mostrar la ecuación para que compruebe si es la ecuación con la que desea trabajar.
- Evaluar la ecuación para un valor de x. El valor debe ser x debe ser introducido por el usuario.
- Imprimir la ecuación polinomial
- Determinar la raíz real de ecuación utilizando el **método de la secante**. Para esto el usuario debe definir el margen de error. La funcionalidad debe mostrar no solo el resultado final sino también los resultados parciales por cada iteración del procedimientos de la siguiente manera:

En la iteración X la raíz hallada es Y.

- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 5. Se tiene una ecuación polinomial de la forma $ax^n+bx^{n-1}+...+yx+z=0$ y se quiere realizar una aplicación en consola utilizando el lenguaje de programación de Java la cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre la ecuación polinomial:

- Definir el grado de la ecuación y cada uno de los coeficientes de la ecuación polinomial. Una vez que el usuario termine esta opción se le debe mostrar la ecuación para que compruebe si es la ecuación con la que desea trabajar.
- Determinar si la ecuación polinomial tiene algún cero dentro de un rango. El rango debe ser definido por el usuario con un limite inferior y un limite superior.
- Cambiar alguno de los coeficiente de la ecuación. Una vez que el usuario termine esta opción se le debe mostrar la ecuación para que compruebe si es la ecuación con la que desea trabajar.
- Evaluar la ecuación para un valor de x. El valor debe ser x debe ser introducido por el usuario.
- Imprimir la ecuación polinomial
- Determinar la raíz real de ecuación utilizando el **método de Newton-Raphson**. Para esto el usuario debe definir el margen de error. La funcionalidad debe mostrar no solo el resultado final sino también los resultados parciales por cada iteración del procedimientos de la siguiente manera:

En la iteración X la raíz hallada es Y.

- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 6. Se tiene una ecuación polinomial de la forma $ax^n+bx^{n-1}+...+yx+z=0$ y se quiere realizar una aplicación en consola utilizando el lenguaje de programación de Java la cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre la ecuación polinomial:

- Definir el grado de la ecuación y cada uno de los coeficientes de la ecuación polinomial. Una vez que el usuario termine esta opción se le debe mostrar la ecuación para que compruebe si es la ecuación con la que desea trabajar.
- Determinar si la ecuación polinomial tiene algún cero dentro de un rango. El rango debe ser definido por el usuario con un limite inferior y un limite superior.
- Cambiar alguno de los coeficiente de la ecuación. Una vez que el usuario termine esta opción se le debe mostrar la ecuación para que compruebe si es la ecuación con la que desea trabajar.
- Evaluar la ecuación para un valor de x. El valor debe ser x debe ser introducido por el usuario.
- Imprimir la ecuación polinomial
- Determinar la raíz real de ecuación utilizando el **método de bisección**. Para esto el usuario debe definir el margen de error. La funcionalidad debe mostrar no solo el resultado final sino también los resultados parciales por cada iteración del procedimientos de la siguiente manera:

En la iteración X el punto medio es Y.

- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 7. Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java el cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre matrices con valores numéricos enteros :

- La suma de dos matrices entrada por el usuario. La aplicación debe validar que dicha operación se puede efectuar de ser posible le debe mostrar el resultado al usuario de lo contrario mostrar un cartel con la explicación de porque no se pudo efectuar la operación.
- Determinar la fila de la matriz donde la suma de sus elementos es mayor. La matriz debe ser proporcionada por el usuario.
- Contar la cantidad de elementos de una matriz que son iguales que un valor. La matriz y el valor deben ser proporcionado por el usuario.
- La división de una matriz por un escalar. Tanto la matriz como el valor escalar son proporcionadas por el usuario.
- Hallar la traspuesta de una matriz. La matriz debe ser proporcionada por el usuario.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 8. Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java el cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre matrices con valores numéricos enteros :

- Definir cuantas operaciones mínimas se deben realizar sobre la matriz para definir la matriz como matriz prima. Una matriz prima es aquella que tiene al menos una fila o columna donde todos sus elementos son números primos. Una operación sobre la matriz es escoger cualquier posición de la matriz y el valor que se encuentra en esa posición incrementarlo en uno. Esta operación solo se puede aplicar sobre matrices cuyos valores no sobrepasen el 10^5 . La matriz debe ser proporcionada por el usuario y se debe validar que la matriz cumpla con la restricción para poder aplicar la operación.
- La multiplicación de una matriz por un escalar. Tanto la matriz como el valor escalar son proporcionadas por el usuario.
- Determinar la columna de la matriz donde la suma de sus elementos es mayor. La matriz debe ser proporcionada por el usuario.
- Contar la cantidad de elementos de una matriz que son mayores que un valor. La matriz y el valor deben ser proporcionado por el usuario.
- Determinar el rango de la matriz. La matriz debe ser proporcionada por el usuario.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 9. Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java el cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre matrices con valores numéricos enteros :

- La multiplicación de dos matrices entrada por el usuario. La aplicación debe validar que dicha operación se puede efectuar de ser posible le debe mostrar el resultado al usuario de lo contrario mostrar un cartel con la explicación de porque no se pudo efectuar la operación.
- Determinar la fila y columna de la matriz donde se encuentra el mayor elemento. La matriz debe ser proporcionada por el usuario.
- Contar la cantidad de elementos de una matriz que son menores que un valor. La matriz y el valor deben ser proporcionado por el usuario.
- Rotar la matriz 90 grados hacia la derecha una X cantidad de veces. Tanto la matriz como la cantidad de rotaciones deben ser proporcionada por el usuario.
- Determinar la inversa de la matriz. La matriz debe ser proporcionada por el usuario.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 10. Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java el cual permita realizar en una sola ejecución del programa un número indeterminado de veces cualquiera de las siguientes operaciones sobre matrices con valores numéricos enteros :

- La resta de dos matrices entrada por el usuario. La aplicación debe validar que dicha operación se puede efectuar de ser posible le debe mostrar el resultado al usuario de lo contrario mostrar un cartel con la explicación de porque no se pudo efectuar la operación.
- Hallar el determinante de una matriz. La matriz debe ser proporcionada por el usuario.
- Rotar la matriz 90 grados hacia la izquierda una X cantidad de veces. Tanto la matriz como la cantidad de rotaciones deben ser proporcionada por el usuario.
- Determinar la fila y columna de la matriz donde se encuentra el menor elemento. La matriz debe ser proporcionada por el usuario.
- Determinar la enésima potencia de una matriz. Tanto la matriz como el valor del exponente de la potencia deben ser proporcionados por el usuario.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 11. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un grafo dirigido no ponderado. Las operaciones son las siguientes:

- Construir un grafo orientado dada su matriz de adyacencia.
- Determinar si en el grafo existe al menos un ciclo.
- Determinar el mayor grado exterior.
- Determinar cuantos nodos tienen un grado exterior superior a un valor proporcionado por el usuario.
- Eliminar y añadir aristas al grafo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 12. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un grafo dirigido no ponderado. Las operaciones son las siguientes:

- Construir un grafo orientado dada su matriz de adyacencia.
- Determinar si en el grafo es un árbol.
- Determinar el mayor grado interior.
- Determinar cuantos nodos tienen un grado interior superior a un valor proporcionado por el usuario.
- Eliminar y añadir aristas al grafo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 13. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un grafo dirigido no ponderado. Las operaciones son las siguientes:

- Construir un grafo orientado dada su matriz de adyacencia.
- Determinar si en el grafo es un árbol torneo.
- Determinar el menor grado interior.
- Determinar cuantos nodos tienen un grado interior inferior a un valor proporcionado por el usuario.
- Eliminar y añadir aristas al grafo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 14. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un grafo dirigido no ponderado. Las operaciones son las siguientes:

- Construir un grafo orientado dada su matriz de adyacencia.
- Determinar si una secuencia de vértices dada es un camino.
- Determinar el menor grado exterior.
- Determinar cuantos nodos tienen un grado exterior inferior a un valor proporcionado por el usuario.
- Eliminar y añadir aristas al grafo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 15. Se tiene la función de la forma $ax^2+bx+c=0$ y desea realizar una aplicación de consola utilizando el lenguaje de programación Java que permita realizar un grupo de operaciones sobre cualquier función de la forma anteriormente mencionada. Dichas operaciones son:

- Permitir al usuario definir los coeficientes de la función
- Imprimir la función
- Evaluar la función para un valor x. Dicho valor x debe ser proporcionado por el usuario.
- Determinar si para un intervalo dado la función tiene raíz o ceros. Dicho intervalo debe ser definido por usuario con el límite inferior y límite superior.
- Determinar para un intervalo del eje x dado el área bajo la curva de la función comprendido para ese intervalo. Para esto debe aplicar **integral de Riemann** todo esto implica que el usuario para esta funcionalidad debe proporcionar un intervalo definido por su límite inferior y superior y la cantidad de particiones con la que se desea realizar el calculo del área.
- Cambiar los valores de los coeficientes de la función.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 16. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un grafo dirigido no ponderado. Las operaciones son las siguientes:

- Construir un grafo orientado dada su matriz de adyacencia.
- Determinar la cantidad de caminos de longitud n que existen entre dos nodos del grafo. El usuario debe decidir cuales son los dos nodos y el valor de n .
- Determinar el nodo de menor sumatoria de su grado interior y exterior .
- Determinar cuantos nodos tienen una sumatoria de grado interior y exterior superior a un valor proporcionado por el usuario.
- Eliminar y añadir aristas al grafo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 17. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de números enteros. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Eliminar un elemento del conjunto dada su posición en el conjunto. La posición debe ser proporcionado por el usuario.
- Imprimir el conjunto ordenado por los valores de forma ascendente sin modificar el conjunto original.
- Dado otro conjunto proporcionado por el usuario determinar el conjunto intersección.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 18. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de números enteros. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Eliminar un elemento del conjunto dado su valor. El valor debe ser proporcionado por el usuario.
- Imprimir el conjunto ordenado por los valores de forma descendente sin modificar el conjunto original.
- Dado otro conjunto proporcionado por el usuario determinar el conjunto diferencia entre el que se tenía menos el proporcionado para esta funcionalidad.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 19. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de números enteros. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Eliminar un elemento del conjunto dado su valor. El valor debe ser proporcionado por el usuario.
- Imprimir el conjunto ordenado tomando como primer criterio de ordenamiento su paridad y como segundo criterio de ordenamiento su valor por los valores de forma descendente sin modificar el conjunto original. Por ejemplo para el conjunto {1,2,3,4,5,6,7,8,9} quedaría de la siguiente manera {8,6,4,2,9,7,5,3,1}
- Dado otro conjunto proporcionado por el usuario determinar el conjunto unión.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 20. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de números enteros. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Insertar un elemento en el conjunto dado su valor y posición. El valor y posición debe ser proporcionado por el usuario. El sistema debe validar que se pueda realizar dicha operación.
- Imprimir el conjunto ordenado tomando como primer criterio de ordenamiento su paridad y como segundo criterio de ordenamiento su valor por los valores de forma ascendente sin modificar el conjunto original. Por ejemplo para el conjunto {1,2,3,4,5,6,7,8,9} quedaría de la siguiente manera {2,4,6,8,1,3,5,7,9}
- Determinar el conjunto potencia.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 21. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de palabras. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Insertar un elemento en el conjunto dado su valor y posición. El valor y posición debe ser proporcionado por el usuario. El sistema debe validar que se pueda realizar dicha operación.
- Hallar la cantidad y las palabras del conjunto que son palabras palíndromos.
- Seleccionar una palabra del conjunto por su posición e invertirla. La posición debe ser proporcionada por el usuario.
- Imprimir el conjunto de palabras.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 22. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de palabras. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Eliminar un elemento en el conjunto dado su posición. La posición debe ser proporcionado por el usuario. El sistema debe validar que se pueda realizar dicha operación.
- Hallar la cantidad y las palabras del conjunto que son anagramas con otra palabra del conjunto.
- Dada letras **x** y **z** debe sustituir en todas las palabras del conjunto las ocurrencias de la letra **x** por la letra **z**. Las letras **x** y **z** deben ser proporcionada por el usuario.
- Imprimir el conjunto de palabras.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 23. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones estadísticas sobre una colección de valores numéricos. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El elemento debe ser proporcionado por el usuario.
- Eliminar un elemento en el conjunto dado su posición. La posición debe ser proporcionado por el usuario. El sistema debe validar que se pueda realizar dicha operación.
- Calcular la moda de la colección .
- Calcular la mediana de la colección.
- Calcular la media aritmética de la colección.
- Imprimir los elementos de la colección.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 24. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de números enteros. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Eliminar un elemento del conjunto dada su valor en el conjunto. El valor debe ser proporcionado por el usuario.
- Dado un valor determinar si dicho valor se encuentra almacenado o no dentro de los elementos del conjunto. El valor debe ser proporcionado por el usuario.
- Dado otro conjunto proporcionado por el usuario determinar el conjunto diferencia simétrica.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 25. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 2D. Las operaciones son la siguientes:

- Crear una colección de puntos 2D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 2D a la colección. El punto 2D debe ser proporcionado por el usuario.
- Eliminar un punto 2D del conjunto dada su posición en la colección. La posición debe ser proporcionada por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los dos puntos mas cercanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de subconjuntos de 3 o más puntos tal que todos los puntos de ese subconjunto sea colineales.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 26. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 2D. Las operaciones son la siguientes:

- Crear una colección de puntos 2D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 2D a la colección. El punto 2D debe ser proporcionado por el usuario.
- Determinar de los puntos almacenados la cantidad de ellos tal que el valor de su abscisa sea igual a un valor proporcionado por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los dos puntos mas lejanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de triadas diferentes tales que los los puntos que pertenezcan a una triada se puedan utilizar como vértices de un triángulo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 27. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 2D. Las operaciones son la siguientes:

- Crear una colección de puntos 2D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 2D a la colección. El punto 2D debe ser proporcionado por el usuario.
- Determinar de los puntos almacenados la cantidad de ellos tal que el valor de su en el eje y sea igual a un valor proporcionado por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los tres puntos mas lejanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de cuartetos diferentes tales que los los puntos que pertenezcan a una cuarteta se puedan utilizar como vértices de un cuadrado.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 28. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 2D. Las operaciones son la siguientes:

- Crear una colección de puntos 2D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 2D a la colección. El punto 2D debe ser proporcionado por el usuario.
- Determinar de los puntos almacenados la cantidad de ellos tal que el valor de su en el eje y sea diferentes a un valor proporcionado por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los tres puntos mas cercanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de cuartetos diferentes tales que los los puntos que pertenezcan a una cuarteta se puedan utilizar como vértices de un rectángulo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 29. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 3D. Las operaciones son la siguientes:

- Crear una colección de puntos 3D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 3D a la colección. El punto 3D debe ser proporcionado por el usuario.
- Eliminar un punto 3D del conjunto dada su posición en la colección. La posición debe ser proporcionada por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los dos puntos mas cercanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de subconjuntos de 3 o más puntos tal que todos los puntos de ese subconjunto sea colineales.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 30. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 3D. Las operaciones son la siguientes:

- Crear una colección de puntos 3D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 3D a la colección. El punto 3D debe ser proporcionado por el usuario.
- Determinar de los puntos almacenados la cantidad de ellos tal que el valor de su abscisa sea igual a un valor proporcionado por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los dos puntos mas lejanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de cuartetos diferentes tales que los los puntos que pertenezcan a una cuarteta se puedan utilizar como vértices de un pirámide de base triangular.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 31. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 3D. Las operaciones son la siguientes:

- Crear una colección de puntos 3D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 3D a la colección. El punto 3D debe ser proporcionado por el usuario.
- Determinar de los puntos almacenados la cantidad de ellos tal que el valor de su en el eje z sea igual a un valor proporcionado por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los tres puntos mas lejanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de subconjuntos de 8 puntos diferentes tales que los los puntos que pertenezcan al subconjunto se puedan utilizar como vértices de un cubo.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 32. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de puntos 2D. Las operaciones son la siguientes:

- Crear una colección de puntos 3D vacío a partir de una capacidad definida por el usuario
- Adicionar un punto 3D a la colección. El punto 3D debe ser proporcionado por el usuario.
- Determinar de los puntos almacenados la cantidad de ellos tal que el valor de su en el eje z sea diferentes a un valor proporcionado por el usuario.
- Determinar de los puntos almacenados en la colección cuales son los tres puntos mas cercanos entre sí .
- Imprimir la colección de puntos.
- Determinar la cantidad de subconjuntos de 8 puntos diferentes tales que los los puntos que pertenezcan a un subconjunto se puedan utilizar como vértices de un prisma de base rectangular.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 33. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de rectas del plano de la forma $Ax+By+C=0$. Las operaciones son la siguientes:

- Crear una colección de rectas vacía a partir de una capacidad definida por el usuario
- Adicionar una recta a la colección. El punto los coeficientes A, B, y C de la ecuación de la recta debe ser proporcionado por el usuario.
- Dado un punto 2D determinar a la cantidad de rectas almacenadas en la colección a los cuales dicho punto pertenece.
- Determinar la cantidad de pares de rectas almacenados en la colección cuales son los paralelas entre sí.
- Imprimir la colección de rectas.
- Determinar de las rectas almacenadas cual es la recta de menor pendiente e imprimir dicha recta con su pendiente. En caso de existir varias imprima la última adicionada a la colección.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 34. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de rectas del plano de la forma $Ax+By+C=0$. Las operaciones son la siguientes:

- Crear una colección de rectas vacío a partir de una capacidad definida por el usuario
- Adicionar una recta a la colección. El punto los coeficientes A, B, y C de la ecuación de la recta debe ser proporcionado por el usuario.
- Eliminar una recta de la colección dada su posición en la misma. La posición debe ser proporcionada por el usuario y el sistema debe validar que se pueda efectuar la operación.
- Determinar la cantidad de pares de rectas almacenados en la colección cuales son los perpendiculares entre sí.
- Imprimir la colección de rectas.
- Determinar de las rectas almacenadas cual es la recta de mayor pendiente e imprimir dicha recta con su pendiente. En caso de existir varias imprima la última adicionada a la colección.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 35. Se desea realizar una aplicación de consola utilizando el lenguaje de programación Java el cual permita realizar un grupo de operaciones sobre una colección de rectas del plano de la forma $Ax+By+C=0$. Las operaciones son la siguientes:

- Crear una colección de rectas vacía a partir de una capacidad definida por el usuario
- Adicionar una recta a la colección. El punto los coeficientes A, B, y C de la ecuación de la recta debe ser proporcionado por el usuario.
- Eliminar una recta de la colección dada su posición en la misma. La posición debe ser proporcionada por el usuario y el sistema debe validar que se pueda efectuar la operación.
- Dada dos rectas almacenadas en la colección determinar su punto de intersección. En caso de no tener punto intersección se le debe informar al usuario a través de un mensaje. Las rectas deben ser seleccionada por usuario el cual debe proporcionar las posiciones donde se encuentran las rectas en la colección.
- Imprimir la colección de rectas.
- Determinar cuantas de las rectas almacenadas tienen una pendiente superior a un valor dado. El valor debe ser proporcionado por el usuario.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

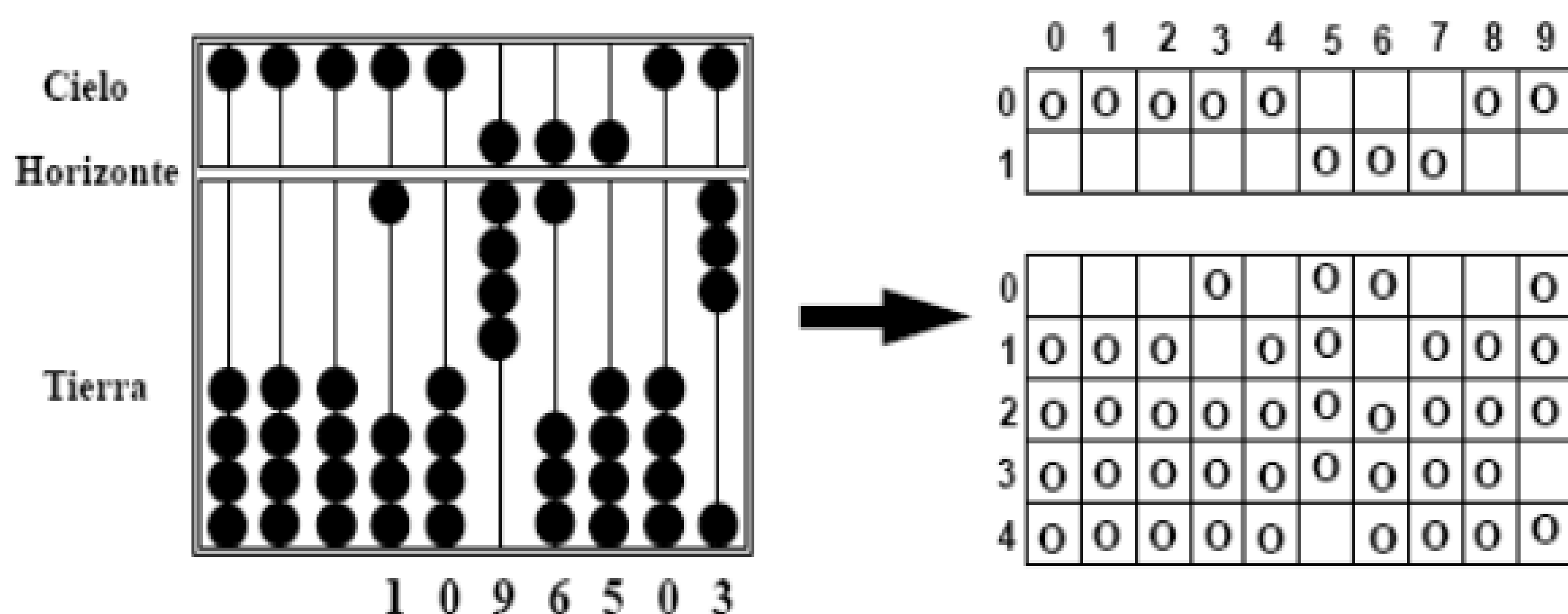
Ejercicio 36. Se desea realizar una aplicación de consola utilizando el lenguaje de programación de Java que permita modelar y realizar un grupo de operaciones sobre un conjunto de números enteros. Las operaciones son las siguientes:

- Crear un conjunto vacío a partir de una capacidad definida por el usuario
- Adicionar un elemento al conjunto. El sistema debe validar que elemento no se encuentra en el conjunto antes de adicionarlo al conjunto. El elemento debe ser proporcionado por el usuario.
- Eliminar un elemento del conjunto dada su posición en el conjunto. La posición debe ser proporcionado por el usuario.
- Dado un valor determinar si dicho valor se encuentra almacenado o no dentro de los elementos del conjunto. El valor debe ser proporcionado por el usuario.
- Dado otro conjunto proporcionado por el usuario determinar el conjunto cartesiano.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.

Ejercicio 37. El soroban es un ábaco japonés que está formado por un marco con cierto número de barras en su interior. Cada barra contiene cierto número de bolas. La disposición de las bolas en cada barra representa un dígito, y todas las barras juntas representan un número: la de más a la derecha para las unidades, la siguiente para las decenas, etc.

Cada barra del soroban está dividida en dos secciones: el contador inferior (Tierra) que consta de 4 bolas, y el contador superior (Cielo) que consta de 1 sola bola. La bola del contador superior (Cielo) equivale a 5 bolas en el contador inferior (Tierra). Las bolas en el contador del Cielo se mueven de arriba hacia abajo, mientras que las bolas en el contador de la Tierra se mueven de abajo hacia arriba, para encontrarse y formar los dígitos en la línea del Horizonte. Por ejemplo, 3 bolas de la Tierra en el Horizonte representan el valor 3, 1 bola del Cielo en el Horizonte representa el valor 5; el dígito representado es la suma de valores de las bolas que están en el Horizonte. La figura muestra la configuración de las bolas en el soroban para representar el número 1.096.503. Se dispone de una versión informática del soroban, capaz de representar números enteros positivos. Para ello se representa el Cielo y la Tierra como dos matrices. La matriz del Cielo consta únicamente de dos filas, mientras que la matriz de la Tierra consta de cinco filas. En ambos casos queda una sola casilla libre en cada columna (ver el ejemplo en la figura).



Se desea realizar una aplicación en consola utilizando el lenguaje de programación Java que permita realizar un grupo de operaciones con el soroban. Dichas operaciones son:

- Crear un Soroban con m barras (m puede variar en cada corrida del programa)
- Dado un número entero representarlo en el soroban.
- Mostrar el número entero que se encuentra representado en el soroban.
- Mostrar gráficamente el soroban.
- Implemente un método que multiplique al soroban por un número entero dado si es posible.
- Diga la cantidad de números que pueden ser representados por el soroban.

- Añada una función suma al soroban que permita sumarlo con otro soroban.
- Terminar la ejecución del programa cuando el usuario lo desee.

La aplicación debe mostrar un menú con las operaciones anteriores para darle la posibilidad al usuario cual el desea hacer.