**RETO 2.1**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 4.4 | | nicolas | m | matematicas | 4.5 | | maria | f | idiomas | 4.3 | | nicolas | m | idiomas | 3.1 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?  ¿Cuántos exámenes tienen una calificación Excelente?  ¿Cuál es la materia con el mejor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia historia?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | historia | 1 | | literatura | 2 | | biologia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 4.6 1.0 0.0 2.0 0.1 1.0 0.0 3.0 2.3 2.0 0.0 1.0 2.5 2.0 0.0 2.0 3.3 2.0 0.0 3.0 0.8 3.0 0.0 1.0 4.5 3.0 0.0 2.0 2.3 3.0 0.0 3.0 3.7 4.0 1.0 1.0 0.7 4.0 1.0 2.0 2.1 4.0 1.0 3.0 4.8 5.0 1.0 1.0 2.0 5.0 1.0 2.0 2.6 5.0 1.0 3.0 1.4 6.0 1.0 1.0 4.7 6.0 1.0 2.0 3.4 6.0 1.0 3.0 1.8 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.2**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 8.6 | | armando | m | matematicas | 9.6 | | maria | f | idiomas | 8.1 | | armando | m | idiomas | 7.3 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?  ¿Cuántos exámenes tienen una calificación Sobresaliente?  ¿Cuál es la materia con el peor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia literatura?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | literatura | 1 | | biologia | 2 | | geografia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 3.8 1.0 0.0 2.0 4.1 1.0 0.0 3.0 2.7 2.0 0.0 1.0 8.9 2.0 0.0 2.0 4.5 2.0 0.0 3.0 8.9 3.0 0.0 1.0 6.9 3.0 0.0 2.0 1.9 3.0 0.0 3.0 9.0 4.0 1.0 1.0 0.4 4.0 1.0 2.0 1.3 4.0 1.0 3.0 9.1 5.0 1.0 1.0 5.0 5.0 1.0 2.0 5.8 5.0 1.0 3.0 2.2 6.0 1.0 1.0 6.1 6.0 1.0 2.0 1.1 6.0 1.0 3.0 8.2 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.3**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 88.4 | | armando | m | matematicas | 97.5 | | marcela | f | idiomas | 83.1 | | armando | m | idiomas | 74.5 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?  ¿Cuántos exámenes tienen una calificación Regular?  ¿Cuál es la materia con el mejor desempeño promedio para el género femenino?  ¿Cuál es el estudiante con el mejor desempeño para la materia biología?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | biologia | 1 | | geografia | 2 | | matemáticas | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 57.3 1.0 0.0 2.0 10.0 1.0 0.0 3.0 20.6 2.0 0.0 1.0 45.1 2.0 0.0 2.0 52.7 2.0 0.0 3.0 65.3 3.0 0.0 1.0 86.9 3.0 0.0 2.0 98.1 3.0 0.0 3.0 97.3 4.0 1.0 1.0 40.2 4.0 1.0 2.0 11.3 4.0 1.0 3.0 94.8 5.0 1.0 1.0 11.1 5.0 1.0 2.0 35.9 5.0 1.0 3.0 46.4 6.0 1.0 1.0 9.3 6.0 1.0 2.0 87.0 6.0 1.0 3.0 2.3 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.4**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 3.9 | | nicolas | m | matematicas | 4.6 | | alexandra | f | idiomas | 3.5 | | nicolas | m | idiomas | 3.0 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?  ¿Cuántos exámenes tienen una calificación insuficiente?  ¿Cuál es la materia con el peor desempeño promedio para el género masculino?  ¿Cuál es el estudiante con el mejor desempeño para la materia geografía?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | geografia | 1 | | matemáticas | 2 | | informatica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 3.0 1.0 0.0 2.0 2.4 1.0 0.0 3.0 4.9 2.0 0.0 1.0 1.6 2.0 0.0 2.0 3.2 2.0 0.0 3.0 0.2 3.0 0.0 1.0 3.1 3.0 0.0 2.0 1.3 3.0 0.0 3.0 0.4 4.0 1.0 1.0 2.1 4.0 1.0 2.0 4.2 4.0 1.0 3.0 0.2 5.0 1.0 1.0 4.4 5.0 1.0 2.0 2.5 5.0 1.0 3.0 2.7 6.0 1.0 1.0 4.3 6.0 1.0 2.0 3.6 6.0 1.0 3.0 2.9 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.5**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 9.0 | | daniel | m | matematicas | 9.7 | | alexandra | f | idiomas | 8.7 | | daniel | m | idiomas | 7.2 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?  ¿Cuántos exámenes tienen una calificación Deficiente?  ¿Qué género tiene un mejor desempeño promedio?  ¿Cuál es el estudiante con el mejor desempeño para la materia matemáticas?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | matemáticas | 1 | | informatica | 2 | | fisica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 5.8 1.0 0.0 2.0 1.9 1.0 0.0 3.0 0.3 2.0 0.0 1.0 0.2 2.0 0.0 2.0 1.1 2.0 0.0 3.0 1.2 3.0 0.0 1.0 2.2 3.0 0.0 2.0 4.8 3.0 0.0 3.0 3.3 4.0 1.0 1.0 5.3 4.0 1.0 2.0 8.1 4.0 1.0 3.0 7.1 5.0 1.0 1.0 0.2 5.0 1.0 2.0 8.1 5.0 1.0 3.0 4.1 6.0 1.0 1.0 0.3 6.0 1.0 2.0 9.9 6.0 1.0 3.0 7.5 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.6**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 82.8 | | daniel | m | matematicas | 93.9 | | alexandra | f | idiomas | 81.4 | | daniel | m | idiomas | 62.3 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?  ¿Qué porcentaje de los exámenes fueron Excelentes?  ¿Cuál es la materia con el mayor número de exámenes aprobados?  ¿Cuál es el estudiante con el mejor desempeño para la materia informática?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | informatica | 1 | | fisica | 2 | | quimica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 91.4 1.0 0.0 2.0 93.8 1.0 0.0 3.0 99.2 2.0 0.0 1.0 17.3 2.0 0.0 2.0 22.1 2.0 0.0 3.0 65.2 3.0 0.0 1.0 5.7 3.0 0.0 2.0 60.2 3.0 0.0 3.0 84.0 4.0 1.0 1.0 28.2 4.0 1.0 2.0 77.0 4.0 1.0 3.0 10.4 5.0 1.0 1.0 71.1 5.0 1.0 2.0 30.9 5.0 1.0 3.0 12.8 6.0 1.0 1.0 54.6 6.0 1.0 2.0 19.0 6.0 1.0 3.0 62.7 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.7**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 4.4 | | daniel | m | matematicas | 4.8 | | marcela | f | idiomas | 4.1 | | daniel | m | idiomas | 2.7 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?  ¿Qué porcentaje de los exámenes fueron Sobresalientes?  ¿Cuál es la materia con el mayor numero de examenes reprobados  ¿Cuál es el estudiante con el mejor desempeño para la materia física?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | fisica | 1 | | quimica | 2 | | idiomas | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 1.3 1.0 0.0 2.0 4.6 1.0 0.0 3.0 3.7 2.0 0.0 1.0 4.1 2.0 0.0 2.0 1.8 2.0 0.0 3.0 0.1 3.0 0.0 1.0 0.5 3.0 0.0 2.0 4.0 3.0 0.0 3.0 2.7 4.0 1.0 1.0 0.3 4.0 1.0 2.0 3.9 4.0 1.0 3.0 4.3 5.0 1.0 1.0 0.8 5.0 1.0 2.0 3.4 5.0 1.0 3.0 3.7 6.0 1.0 1.0 0.6 6.0 1.0 2.0 4.3 6.0 1.0 3.0 1.3 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.8**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 8.2 | | armando | m | matematicas | 9.6 | | maria | f | idiomas | 8.6 | | armando | m | idiomas | 7.2 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el desempeño promedio de todo el grupo?  ¿Cuántos exámenes tienen una calificación Excelente?  ¿Cuál es la materia con el peor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia biología?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | biologia | 1 | | geografia | 2 | | matemáticas | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 3.9 1.0 0.0 2.0 8.2 1.0 0.0 3.0 8.4 2.0 0.0 1.0 6.4 2.0 0.0 2.0 5.5 2.0 0.0 3.0 4.1 3.0 0.0 1.0 7.1 3.0 0.0 2.0 4.9 3.0 0.0 3.0 0.3 4.0 1.0 1.0 4.6 4.0 1.0 2.0 2.1 4.0 1.0 3.0 3.1 5.0 1.0 1.0 0.6 5.0 1.0 2.0 0.2 5.0 1.0 3.0 9.5 6.0 1.0 1.0 4.3 6.0 1.0 2.0 9.8 6.0 1.0 3.0 8.8 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.9**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 87.6 | | nicolas | m | matematicas | 99.6 | | maria | f | idiomas | 85.9 | | nicolas | m | idiomas | 77.9 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el desempeño promedio de todo el grupo?  ¿Cuántos exámenes tienen una calificación Sobresaliente?  ¿Cuál es la materia con el mejor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia geografía?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | geografia | 1 | | matemáticas | 2 | | informatica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 94.6 1.0 0.0 2.0 70.7 1.0 0.0 3.0 85.5 2.0 0.0 1.0 46.6 2.0 0.0 2.0 64.5 2.0 0.0 3.0 71.0 3.0 0.0 1.0 0.2 3.0 0.0 2.0 58.2 3.0 0.0 3.0 19.6 4.0 1.0 1.0 73.3 4.0 1.0 2.0 15.3 4.0 1.0 3.0 62.1 5.0 1.0 1.0 37.2 5.0 1.0 2.0 65.2 5.0 1.0 3.0 0.5 6.0 1.0 1.0 20.3 6.0 1.0 2.0 81.2 6.0 1.0 3.0 86.2 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.10**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 4.3 | | nicolas | m | matematicas | 4.8 | | maria | f | idiomas | 4.0 | | nicolas | m | idiomas | 3.0 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el desempeño promedio de todo el grupo?  ¿Cuántos exámenes tienen una calificación Regular?  ¿Cuál es la materia con el peor desempeño promedio para el género masculino?  ¿Cuál es el estudiante con el mejor desempeño para la materia historia?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | historia | 1 | | literatura | 2 | | biologia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 3.0 1.0 0.0 2.0 2.8 1.0 0.0 3.0 3.6 2.0 0.0 1.0 2.8 2.0 0.0 2.0 1.2 2.0 0.0 3.0 2.8 3.0 0.0 1.0 3.0 3.0 0.0 2.0 3.5 3.0 0.0 3.0 0.9 4.0 1.0 1.0 0.3 4.0 1.0 2.0 4.9 4.0 1.0 3.0 0.2 5.0 1.0 1.0 1.4 5.0 1.0 2.0 1.9 5.0 1.0 3.0 4.1 6.0 1.0 1.0 4.0 6.0 1.0 2.0 0.7 6.0 1.0 3.0 0.4 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.11**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 8.4 | | nicolas | m | matematicas | 9.1 | | maria | f | idiomas | 8.2 | | nicolas | m | idiomas | 7.9 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el desempeño promedio de todo el grupo?  ¿Cuántos exámenes tienen una calificación Insuficiente?  ¿Cuál es la materia con el mejor desempeño promedio para el género femenino?  ¿Cuál es el estudiante con el mejor desempeño para la materia literatura?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | literatura | 1 | | biologia | 2 | | geografia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 2.4 1.0 0.0 2.0 8.9 1.0 0.0 3.0 6.2 2.0 0.0 1.0 6.4 2.0 0.0 2.0 3.8 2.0 0.0 3.0 0.2 3.0 0.0 1.0 3.7 3.0 0.0 2.0 0.5 3.0 0.0 3.0 8.2 4.0 1.0 1.0 7.2 4.0 1.0 2.0 4.2 4.0 1.0 3.0 8.2 5.0 1.0 1.0 5.4 5.0 1.0 2.0 0.1 5.0 1.0 3.0 4.9 6.0 1.0 1.0 1.0 6.0 1.0 2.0 2.4 6.0 1.0 3.0 2.7 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.12**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 88.6 | | armando | m | matematicas | 91.0 | | alexandra | f | idiomas | 87.9 | | armando | m | idiomas | 74.1 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el desempeño promedio de todo el grupo?  ¿Cuántos exámenes tienen una calificación Deficiente?  ¿Cuál es la materia con el mayor número de exámenes aprobados?  ¿Cuál es el estudiante con el mejor desempeño para la materia física?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | fisica | 1 | | quimica | 2 | | idiomas | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 62.1 1.0 0.0 2.0 98.5 1.0 0.0 3.0 88.6 2.0 0.0 1.0 12.4 2.0 0.0 2.0 92.2 2.0 0.0 3.0 60.5 3.0 0.0 1.0 95.2 3.0 0.0 2.0 74.5 3.0 0.0 3.0 28.2 4.0 1.0 1.0 69.0 4.0 1.0 2.0 36.9 4.0 1.0 3.0 73.4 5.0 1.0 1.0 88.0 5.0 1.0 2.0 88.3 5.0 1.0 3.0 67.5 6.0 1.0 1.0 15.8 6.0 1.0 2.0 91.4 6.0 1.0 3.0 77.4 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.13**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 4.2 | | nicolas | m | matematicas | 4.9 | | maria | f | idiomas | 4.4 | | nicolas | m | idiomas | 3.5 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el desempeño promedio de todo el grupo?  ¿Qué porcentaje de los exámenes fueron Excelentes?  ¿Qué género tiene un mejor desempeño promedio?  ¿Cuál es el estudiante con el mejor desempeño para la materia química?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | quimica | 1 | | idiomas | 2 | | historia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 1.9 1.0 0.0 2.0 0.6 1.0 0.0 3.0 4.4 2.0 0.0 1.0 4.1 2.0 0.0 2.0 2.2 2.0 0.0 3.0 4.4 3.0 0.0 1.0 2.2 3.0 0.0 2.0 3.3 3.0 0.0 3.0 0.9 4.0 1.0 1.0 3.1 4.0 1.0 2.0 1.4 4.0 1.0 3.0 4.1 5.0 1.0 1.0 4.5 5.0 1.0 2.0 3.5 5.0 1.0 3.0 4.0 6.0 1.0 1.0 0.4 6.0 1.0 2.0 3.2 6.0 1.0 3.0 4.6 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.14**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 8.6 | | nicolas | m | matematicas | 9.0 | | alexandra | f | idiomas | 8.5 | | nicolas | m | idiomas | 7.1 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es el desempeño promedio de todo el grupo?  ¿Qué porcentaje de los exámenes fueron Regulares?  ¿Cuál es la materia con el mayor numero de examenes reprobados  ¿Cuál es el estudiante con el mejor desempeño para la materia matemáticas?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | matemáticas | 1 | | informatica | 2 | | fisica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 3.0 1.0 0.0 2.0 2.8 1.0 0.0 3.0 0.7 2.0 0.0 1.0 4.5 2.0 0.0 2.0 7.5 2.0 0.0 3.0 2.2 3.0 0.0 1.0 6.5 3.0 0.0 2.0 6.7 3.0 0.0 3.0 3.6 4.0 1.0 1.0 5.7 4.0 1.0 2.0 7.8 4.0 1.0 3.0 8.5 5.0 1.0 1.0 3.2 5.0 1.0 2.0 2.8 5.0 1.0 3.0 0.0 6.0 1.0 1.0 0.6 6.0 1.0 2.0 3.7 6.0 1.0 3.0 5.6 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.15**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 82.2 | | armando | m | matematicas | 94.2 | | maria | f | idiomas | 80.3 | | armando | m | idiomas | 78.3 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es la varianza de las calificaciones para todo el grupo?  ¿Cuántos exámenes tienen una calificación Excelente?  ¿Cuál es la materia con el mejor desempeño promedio para el género femenino?  ¿Cuál es el estudiante con el mejor desempeño para la materia geografía?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | geografia | 1 | | matemáticas | 2 | | informatica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 59.3 1.0 0.0 2.0 43.8 1.0 0.0 3.0 62.3 2.0 0.0 1.0 94.3 2.0 0.0 2.0 45.4 2.0 0.0 3.0 53.8 3.0 0.0 1.0 83.6 3.0 0.0 2.0 57.2 3.0 0.0 3.0 25.3 4.0 1.0 1.0 12.2 4.0 1.0 2.0 67.8 4.0 1.0 3.0 36.2 5.0 1.0 1.0 16.5 5.0 1.0 2.0 5.4 5.0 1.0 3.0 2.3 6.0 1.0 1.0 3.5 6.0 1.0 2.0 3.7 6.0 1.0 3.0 83.1 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.16**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 3.7 | | daniel | m | matematicas | 4.8 | | marcela | f | idiomas | 3.5 | | daniel | m | idiomas | 3.4 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es la varianza de las calificaciones para todo el grupo?  ¿Cuántos exámenes tienen una calificación Sobresaliente?  ¿Cuál es la materia con el peor desempeño promedio para el género masculino?  ¿Cuál es el estudiante con el mejor desempeño para la materia biología?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | biologia | 1 | | geografia | 2 | | matemáticas | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 4.2 1.0 0.0 2.0 3.4 1.0 0.0 3.0 1.3 2.0 0.0 1.0 3.7 2.0 0.0 2.0 1.3 2.0 0.0 3.0 2.1 3.0 0.0 1.0 0.8 3.0 0.0 2.0 2.0 3.0 0.0 3.0 4.6 4.0 1.0 1.0 0.1 4.0 1.0 2.0 4.1 4.0 1.0 3.0 3.1 5.0 1.0 1.0 2.8 5.0 1.0 2.0 2.1 5.0 1.0 3.0 0.5 6.0 1.0 1.0 0.8 6.0 1.0 2.0 1.9 6.0 1.0 3.0 3.9 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.17**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 8.3 | | daniel | m | matematicas | 9.1 | | alexandra | f | idiomas | 8.0 | | daniel | m | idiomas | 6.2 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es la varianza de las calificaciones para todo el grupo?  ¿Cuántos exámenes tienen una calificación Regular?  ¿Cuál es la materia con el mejor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia literatura?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | literatura | 1 | | biologia | 2 | | geografia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 0.7 1.0 0.0 2.0 5.3 1.0 0.0 3.0 8.8 2.0 0.0 1.0 5.2 2.0 0.0 2.0 8.0 2.0 0.0 3.0 5.0 3.0 0.0 1.0 7.3 3.0 0.0 2.0 6.4 3.0 0.0 3.0 4.2 4.0 1.0 1.0 8.6 4.0 1.0 2.0 1.5 4.0 1.0 3.0 9.2 5.0 1.0 1.0 2.4 5.0 1.0 2.0 5.0 5.0 1.0 3.0 9.9 6.0 1.0 1.0 7.0 6.0 1.0 2.0 3.5 6.0 1.0 3.0 1.1 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.18**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 86.1 | | nicolas | m | matematicas | 90.6 | | maria | f | idiomas | 81.4 | | nicolas | m | idiomas | 78.5 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es la varianza de las calificaciones para todo el grupo?  ¿Cuántos exámenes tienen una calificación insuficiente?  ¿Cuál es la materia con el peor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia historia?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | historia | 1 | | literatura | 2 | | biologia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 56.6 1.0 0.0 2.0 1.3 1.0 0.0 3.0 31.4 2.0 0.0 1.0 77.1 2.0 0.0 2.0 17.2 2.0 0.0 3.0 59.8 3.0 0.0 1.0 60.8 3.0 0.0 2.0 9.6 3.0 0.0 3.0 14.6 4.0 1.0 1.0 12.8 4.0 1.0 2.0 70.2 4.0 1.0 3.0 25.6 5.0 1.0 1.0 7.1 5.0 1.0 2.0 92.0 5.0 1.0 3.0 72.6 6.0 1.0 1.0 26.7 6.0 1.0 2.0 98.1 6.0 1.0 3.0 97.2 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.19**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 3.6 | | daniel | m | matematicas | 4.6 | | marcela | f | idiomas | 3.7 | | daniel | m | idiomas | 2.6 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es la varianza de las calificaciones para todo el grupo?  ¿Cuántos exámenes tienen una calificación Deficiente?  ¿Cuál es la materia con el mayor numero de examenes reprobados  ¿Cuál es el estudiante con el mejor desempeño para la materia informática?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | informatica | 1 | | fisica | 2 | | quimica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 0.0 1.0 0.0 2.0 1.3 1.0 0.0 3.0 3.8 2.0 0.0 1.0 1.2 2.0 0.0 2.0 1.0 2.0 0.0 3.0 1.4 3.0 0.0 1.0 0.9 3.0 0.0 2.0 4.8 3.0 0.0 3.0 1.3 4.0 1.0 1.0 4.6 4.0 1.0 2.0 3.6 4.0 1.0 3.0 0.9 5.0 1.0 1.0 3.9 5.0 1.0 2.0 3.9 5.0 1.0 3.0 3.5 6.0 1.0 1.0 2.5 6.0 1.0 2.0 4.7 6.0 1.0 3.0 4.2 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.20**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 8.9 | | armando | m | matematicas | 9.2 | | maria | f | idiomas | 8.8 | | armando | m | idiomas | 7.3 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es la varianza de las calificaciones para todo el grupo?  ¿Qué porcentaje de los exámenes fueron Sobresalientes?  ¿Cuál género tiene un mejor desempeño promedio?  ¿Cuál es el estudiante con el mejor desempeño para la materia idiomas?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | idiomas | 1 | | historia | 2 | | literatura | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 7.7 1.0 0.0 2.0 1.2 1.0 0.0 3.0 1.4 2.0 0.0 1.0 7.7 2.0 0.0 2.0 8.1 2.0 0.0 3.0 0.1 3.0 0.0 1.0 5.5 3.0 0.0 2.0 9.6 3.0 0.0 3.0 7.3 4.0 1.0 1.0 4.0 4.0 1.0 2.0 4.6 4.0 1.0 3.0 5.1 5.0 1.0 1.0 5.7 5.0 1.0 2.0 1.0 5.0 1.0 3.0 7.7 6.0 1.0 1.0 8.1 6.0 1.0 2.0 0.3 6.0 1.0 3.0 6.9 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.21**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 80.7 | | daniel | m | matematicas | 93.9 | | maria | f | idiomas | 80.5 | | daniel | m | idiomas | 61.0 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuál es la varianza de las calificaciones para todo el grupo?  ¿Qué porcentaje de los exámenes fueron Regulares?  ¿Cuál es la materia con el mayor número de exámenes aprobados?  ¿Cuál es el estudiante con el mejor desempeño para la materia química?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | quimica | 1 | | idiomas | 2 | | historia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 48.4 1.0 0.0 2.0 52.8 1.0 0.0 3.0 9.0 2.0 0.0 1.0 48.4 2.0 0.0 2.0 94.1 2.0 0.0 3.0 72.7 3.0 0.0 1.0 64.6 3.0 0.0 2.0 77.5 3.0 0.0 3.0 6.2 4.0 1.0 1.0 36.9 4.0 1.0 2.0 87.0 4.0 1.0 3.0 29.9 5.0 1.0 1.0 48.3 5.0 1.0 2.0 15.5 5.0 1.0 3.0 35.2 6.0 1.0 1.0 13.0 6.0 1.0 2.0 41.2 6.0 1.0 3.0 73.5 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.22**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 4.4 | | armando | m | matematicas | 4.7 | | alexandra | f | idiomas | 4.1 | | armando | m | idiomas | 2.9 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos exámenes tienen una calificación menor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación Excelente?  ¿Cuál es la materia con el peor desempeño promedio para el género masculino?  ¿Cuál es el estudiante con el mejor desempeño para la materia literatura?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | literatura | 1 | | biologia | 2 | | geografia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 0.9 1.0 0.0 2.0 2.4 1.0 0.0 3.0 2.8 2.0 0.0 1.0 1.9 2.0 0.0 2.0 4.3 2.0 0.0 3.0 2.9 3.0 0.0 1.0 0.0 3.0 0.0 2.0 4.5 3.0 0.0 3.0 0.5 4.0 1.0 1.0 2.0 4.0 1.0 2.0 4.7 4.0 1.0 3.0 0.7 5.0 1.0 1.0 1.0 5.0 1.0 2.0 1.9 5.0 1.0 3.0 0.5 6.0 1.0 1.0 1.1 6.0 1.0 2.0 3.0 6.0 1.0 3.0 1.0 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.23**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 8.2 | | daniel | m | matematicas | 9.3 | | alexandra | f | idiomas | 8.4 | | daniel | m | idiomas | 6.6 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos exámenes tienen una calificación menor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación Sobresaliente?  ¿Cuál es la materia con el mejor desempeño promedio para el género femenino?  ¿Cuál es el estudiante con el mejor desempeño para la materia historia?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | historia | 1 | | literatura | 2 | | biologia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 6.0 1.0 0.0 2.0 2.1 1.0 0.0 3.0 1.9 2.0 0.0 1.0 1.6 2.0 0.0 2.0 5.6 2.0 0.0 3.0 4.2 3.0 0.0 1.0 7.0 3.0 0.0 2.0 0.1 3.0 0.0 3.0 0.6 4.0 1.0 1.0 6.9 4.0 1.0 2.0 6.0 4.0 1.0 3.0 10.0 5.0 1.0 1.0 9.0 5.0 1.0 2.0 3.5 5.0 1.0 3.0 3.2 6.0 1.0 1.0 1.8 6.0 1.0 2.0 8.2 6.0 1.0 3.0 7.8 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.24**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 86.1 | | nicolas | m | matematicas | 94.3 | | marcela | f | idiomas | 80.5 | | nicolas | m | idiomas | 77.8 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos exámenes tienen una calificación menor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación Regular?  ¿Cuál es la materia con el peor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia geografía?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | geografia | 1 | | matemáticas | 2 | | informatica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 36.8 1.0 0.0 2.0 52.0 1.0 0.0 3.0 57.5 2.0 0.0 1.0 63.7 2.0 0.0 2.0 55.7 2.0 0.0 3.0 9.2 3.0 0.0 1.0 35.4 3.0 0.0 2.0 52.5 3.0 0.0 3.0 59.2 4.0 1.0 1.0 46.8 4.0 1.0 2.0 14.9 4.0 1.0 3.0 99.0 5.0 1.0 1.0 18.1 5.0 1.0 2.0 31.4 5.0 1.0 3.0 36.0 6.0 1.0 1.0 28.5 6.0 1.0 2.0 51.5 6.0 1.0 3.0 44.6 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.25**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 3.7 | | armando | m | matematicas | 4.6 | | marcela | f | idiomas | 3.7 | | armando | m | idiomas | 2.5 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos exámenes tienen una calificación menor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación insuficiente?  ¿Cuál es la materia con el mejor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia biología?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | biologia | 1 | | geografia | 2 | | matemáticas | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 3.4 1.0 0.0 2.0 0.2 1.0 0.0 3.0 1.6 2.0 0.0 1.0 4.9 2.0 0.0 2.0 4.0 2.0 0.0 3.0 2.1 3.0 0.0 1.0 1.1 3.0 0.0 2.0 3.2 3.0 0.0 3.0 2.3 4.0 1.0 1.0 0.6 4.0 1.0 2.0 0.6 4.0 1.0 3.0 1.9 5.0 1.0 1.0 4.8 5.0 1.0 2.0 2.1 5.0 1.0 3.0 0.3 6.0 1.0 1.0 1.3 6.0 1.0 2.0 2.9 6.0 1.0 3.0 1.6 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.26**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 8.5 | | daniel | m | matematicas | 9.8 | | alexandra | f | idiomas | 8.2 | | daniel | m | idiomas | 6.3 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos exámenes tienen una calificación menor a la del promedio del grupo?  ¿Qué porcentaje de los exámenes fueron Excelentes?  ¿Cuál es la materia con el mayor numero de examenes reprobados  ¿Cuál es el estudiante con el mejor desempeño para la materia idiomas?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | idiomas | 1 | | historia | 2 | | literatura | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 2.1 1.0 0.0 2.0 2.2 1.0 0.0 3.0 2.8 2.0 0.0 1.0 4.6 2.0 0.0 2.0 7.0 2.0 0.0 3.0 7.4 3.0 0.0 1.0 4.1 3.0 0.0 2.0 5.7 3.0 0.0 3.0 3.5 4.0 1.0 1.0 2.2 4.0 1.0 2.0 2.9 4.0 1.0 3.0 7.5 5.0 1.0 1.0 2.5 5.0 1.0 2.0 9.0 5.0 1.0 3.0 9.3 6.0 1.0 1.0 7.9 6.0 1.0 2.0 1.1 6.0 1.0 3.0 3.6 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.27**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 88.3 | | nicolas | m | matematicas | 91.5 | | marcela | f | idiomas | 86.9 | | nicolas | m | idiomas | 67.3 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos exámenes tienen una calificación menor a la del promedio del grupo?  ¿Qué porcentaje de los exámenes fueron Sobresalientes?  ¿Cuál es la materia con el mayor número de exámenes aprobados?  ¿Cuál es el estudiante con el mejor desempeño para la materia matemáticas?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | matemáticas | 1 | | informatica | 2 | | fisica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 66.8 1.0 0.0 2.0 3.8 1.0 0.0 3.0 59.2 2.0 0.0 1.0 25.4 2.0 0.0 2.0 68.1 2.0 0.0 3.0 39.7 3.0 0.0 1.0 12.3 3.0 0.0 2.0 0.1 3.0 0.0 3.0 37.6 4.0 1.0 1.0 51.2 4.0 1.0 2.0 22.0 4.0 1.0 3.0 37.9 5.0 1.0 1.0 99.8 5.0 1.0 2.0 19.5 5.0 1.0 3.0 33.6 6.0 1.0 1.0 23.6 6.0 1.0 2.0 36.6 6.0 1.0 3.0 9.2 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.28**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 4.4 | | daniel | m | matematicas | 4.8 | | maria | f | idiomas | 4.2 | | daniel | m | idiomas | 3.2 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos exámenes tienen una calificación menor a la del promedio del grupo?  ¿Qué porcentaje de los exámenes fueron Regulares?  ¿Cuál género tiene un mejor desempeño promedio?  ¿Cuál es el estudiante con el mejor desempeño para la materia informática?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | informatica | 1 | | fisica | 2 | | quimica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 2.3 1.0 0.0 2.0 2.4 1.0 0.0 3.0 4.8 2.0 0.0 1.0 3.9 2.0 0.0 2.0 2.9 2.0 0.0 3.0 4.3 3.0 0.0 1.0 4.0 3.0 0.0 2.0 0.5 3.0 0.0 3.0 2.5 4.0 1.0 1.0 0.9 4.0 1.0 2.0 0.0 4.0 1.0 3.0 1.8 5.0 1.0 1.0 2.9 5.0 1.0 2.0 0.3 5.0 1.0 3.0 3.1 6.0 1.0 1.0 1.0 6.0 1.0 2.0 2.3 6.0 1.0 3.0 2.2 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.29**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 8.4 | | armando | m | matematicas | 9.7 | | marcela | f | idiomas | 8.7 | | armando | m | idiomas | 7.5 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación Excelente?  ¿Cuál género tiene un mejor desempeño promedio?  ¿Cuál es el estudiante con el mejor desempeño para la materia física?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | fisica | 1 | | quimica | 2 | | idiomas | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 1.9 1.0 0.0 2.0 3.4 1.0 0.0 3.0 2.2 2.0 0.0 1.0 5.6 2.0 0.0 2.0 0.2 2.0 0.0 3.0 7.0 3.0 0.0 1.0 5.3 3.0 0.0 2.0 9.3 3.0 0.0 3.0 3.8 4.0 1.0 1.0 2.3 4.0 1.0 2.0 7.5 4.0 1.0 3.0 3.7 5.0 1.0 1.0 2.2 5.0 1.0 2.0 0.2 5.0 1.0 3.0 0.2 6.0 1.0 1.0 0.3 6.0 1.0 2.0 5.8 6.0 1.0 3.0 2.2 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.30**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 84.3 | | nicolas | m | matematicas | 99.2 | | maria | f | idiomas | 83.0 | | nicolas | m | idiomas | 68.2 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación Sobresaliente?  ¿Cuál es la materia con el mayor número de exámenes aprobados?  ¿Cuál es el estudiante con el mejor desempeño para la materia idiomas?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | idiomas | 1 | | historia | 2 | | literatura | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 77.3 1.0 0.0 2.0 99.7 1.0 0.0 3.0 84.9 2.0 0.0 1.0 45.3 2.0 0.0 2.0 57.0 2.0 0.0 3.0 43.1 3.0 0.0 1.0 24.8 3.0 0.0 2.0 73.4 3.0 0.0 3.0 25.9 4.0 1.0 1.0 75.8 4.0 1.0 2.0 26.0 4.0 1.0 3.0 44.8 5.0 1.0 1.0 39.5 5.0 1.0 2.0 17.8 5.0 1.0 3.0 53.7 6.0 1.0 1.0 15.6 6.0 1.0 2.0 19.2 6.0 1.0 3.0 51.6 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.31**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 4.5 | | armando | m | matematicas | 4.8 | | maria | f | idiomas | 4.2 | | armando | m | idiomas | 3.3 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación Regular?  ¿Cuál es la materia con el mayor numero de examenes reprobados  ¿Cuál es el estudiante con el mejor desempeño para la materia química?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | quimica | 1 | | idiomas | 2 | | historia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 1.2 1.0 0.0 2.0 1.8 1.0 0.0 3.0 2.2 2.0 0.0 1.0 3.4 2.0 0.0 2.0 3.0 2.0 0.0 3.0 4.9 3.0 0.0 1.0 3.6 3.0 0.0 2.0 3.5 3.0 0.0 3.0 4.3 4.0 1.0 1.0 3.5 4.0 1.0 2.0 2.7 4.0 1.0 3.0 0.1 5.0 1.0 1.0 3.6 5.0 1.0 2.0 3.5 5.0 1.0 3.0 3.5 6.0 1.0 1.0 3.7 6.0 1.0 2.0 1.7 6.0 1.0 3.0 3.9 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.32**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 8.5 | | daniel | m | matematicas | 9.6 | | marcela | f | idiomas | 8.2 | | daniel | m | idiomas | 6.8 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo?  ¿Cuántos exámenes tienen una calificación Deficiente?  ¿Cuál es la materia con el mejor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el peor desempeño para la materia historia?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | historia | 1 | | literatura | 2 | | biologia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 0.6 1.0 0.0 2.0 5.0 1.0 0.0 3.0 9.3 2.0 0.0 1.0 3.7 2.0 0.0 2.0 2.3 2.0 0.0 3.0 1.3 3.0 0.0 1.0 9.1 3.0 0.0 2.0 5.6 3.0 0.0 3.0 0.7 4.0 1.0 1.0 3.8 4.0 1.0 2.0 2.8 4.0 1.0 3.0 1.0 5.0 1.0 1.0 6.9 5.0 1.0 2.0 8.0 5.0 1.0 3.0 5.8 6.0 1.0 1.0 6.1 6.0 1.0 2.0 6.2 6.0 1.0 3.0 8.0 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.33**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | maria | f | matematicas | 89.7 | | armando | m | matematicas | 96.4 | | maria | f | idiomas | 86.7 | | armando | m | idiomas | 66.5 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo?  ¿Qué porcentaje de los exámenes fueron Excelentes?  ¿Cuál es la materia con el peor desempeño promedio para todo el grupo?  ¿Cuál es el estudiante con el mejor desempeño para la materia matemáticas?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | matemáticas | 1 | | informatica | 2 | | fisica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 74.9 1.0 0.0 2.0 63.7 1.0 0.0 3.0 57.6 2.0 0.0 1.0 56.5 2.0 0.0 2.0 15.3 2.0 0.0 3.0 20.6 3.0 0.0 1.0 30.1 3.0 0.0 2.0 47.8 3.0 0.0 3.0 67.1 4.0 1.0 1.0 71.3 4.0 1.0 2.0 19.8 4.0 1.0 3.0 9.7 5.0 1.0 1.0 81.6 5.0 1.0 2.0 35.0 5.0 1.0 3.0 65.1 6.0 1.0 1.0 7.9 6.0 1.0 2.0 31.0 6.0 1.0 3.0 46.5 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.34**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 4.1 | | armando | m | matematicas | 4.9 | | marcela | f | idiomas | 4.5 | | armando | m | idiomas | 3.5 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (4.5 - 5] | Excelente | | (3.5 - 4.5] | Sobresaliente | | (2.5 - 3.5] | Regular | | (1 - 2.5] | Insuficiente | | [0 - 1] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo?  ¿Qué porcentaje de los exámenes fueron Sobresalientes?  ¿Cuál es la materia con el mejor desempeño promedio para el género femenino?  ¿Cuál es el estudiante con el mejor desempeño para la materia informática?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | informatica | 1 | | fisica | 2 | | quimica | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 0.3 1.0 0.0 2.0 2.8 1.0 0.0 3.0 2.7 2.0 0.0 1.0 2.1 2.0 0.0 2.0 4.4 2.0 0.0 3.0 4.1 3.0 0.0 1.0 3.2 3.0 0.0 2.0 2.3 3.0 0.0 3.0 3.4 4.0 1.0 1.0 4.9 4.0 1.0 2.0 0.1 4.0 1.0 3.0 2.6 5.0 1.0 1.0 2.4 5.0 1.0 2.0 5.0 5.0 1.0 3.0 2.9 6.0 1.0 1.0 0.2 6.0 1.0 2.0 5.0 6.0 1.0 3.0 1.5 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.35**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 10, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | marcela | f | matematicas | 8.6 | | nicolas | m | matematicas | 9.1 | | marcela | f | idiomas | 8.2 | | nicolas | m | idiomas | 7.8 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (9 - 10] | Excelente | | (8 - 9] | Sobresaliente | | (6 - 8] | Regular | | (3 - 6] | Insuficiente | | [0 - 3] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo?  ¿Qué porcentaje de los exámenes fueron Regulares?  ¿Cuál es la materia con el peor desempeño promedio para el género masculino?  ¿Cuál es el estudiante con el peor desempeño para la materia literatura?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | literatura | 1 | | biologia | 2 | | geografia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 1.4 1.0 0.0 2.0 4.6 1.0 0.0 3.0 1.2 2.0 0.0 1.0 3.6 2.0 0.0 2.0 6.4 2.0 0.0 3.0 7.5 3.0 0.0 1.0 8.7 3.0 0.0 2.0 1.6 3.0 0.0 3.0 5.3 4.0 1.0 1.0 8.6 4.0 1.0 2.0 7.0 4.0 1.0 3.0 1.8 5.0 1.0 1.0 0.5 5.0 1.0 2.0 8.0 5.0 1.0 3.0 0.1 6.0 1.0 1.0 6.0 6.0 1.0 2.0 0.0 6.0 1.0 3.0 8.8 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.36**

|  |  |
| --- | --- |
| Nombre del reto: | Diseño de sistema estadístico para escuela utilizando POO |
| Autor reto: | Carlos Andres Conrado A. |
| Descripción del reto con su respectiva solución: | |
| Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.  La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 100, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.  El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:   |  |  |  |  | | --- | --- | --- | --- | | **Nombre** | **Género** | **Materia** | **Nota** | | alexandra | f | matematicas | 85.9 | | daniel | m | matematicas | 93.9 | | alexandra | f | idiomas | 89.8 | | daniel | m | idiomas | 74.7 |   Las calificaciones en la escuela se asignan con la siguiente escala de rangos:   |  |  | | --- | --- | | Rango de notas | Calificación | | (90 - 100] | Excelente | | (80 - 90] | Sobresaliente | | (60 - 80] | Regular | | (30 - 60] | Insuficiente | | [0 - 30] | Deficiente |   El algoritmo debe ser capaz de responder a las siguientes preguntas:  ¿Cuántos de los exámenes fueron reprobados?  ¿Cuántos exámenes tienen una calificación Excelente?  ¿Cuál es la materia con el mayor número de exámenes aprobados?  ¿Cuál es el estudiante con el peor desempeño para la materia historia?    **EJEMPLO**  Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:   |  |  | | --- | --- | | Nombre | Identificador | | armando | 1 | | nicolas | 2 | | daniel | 3 | | maria | 4 | | marcela | 5 | | alexandra | 6 |  |  |  | | --- | --- | | Materia | Identificador | | historia | 1 | | literatura | 2 | | biologia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 79.1 1.0 0.0 2.0 33.0 1.0 0.0 3.0 36.6 2.0 0.0 1.0 55.3 2.0 0.0 2.0 44.1 2.0 0.0 3.0 13.3 3.0 0.0 1.0 97.6 3.0 0.0 2.0 4.9 3.0 0.0 3.0 66.2 4.0 1.0 1.0 10.6 4.0 1.0 2.0 29.5 4.0 1.0 3.0 66.5 5.0 1.0 1.0 16.0 5.0 1.0 2.0 8.5 5.0 1.0 3.0 26.4 6.0 1.0 1.0 52.8 6.0 1.0 2.0 92.4 6.0 1.0 3.0 77.3 | | Salida del programa | | {{exaple\_output}} |     Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.  **FASE 2**  El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:  Se debe implementar todo en una clase llamada **SchoolGradingSystem**.  La clase debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.  La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).  Utilizar las buenas prácticas de programación.  Notas:  Es importante seleccionar los valores adecuados para cada método, ya que esto puede generar errores a la hora de calificar.  Prestar especial cuidado a las notaciones de los rangos.  Se considera aprobado a una calificación igual o mayor a Regular.  Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.  Al imprimir números decimales estos deben ser redondeados a dos cifras significativas, ejemplo: 3.666666667 → 3.67.  Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.  El archivo debe llamarse **reto1.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**RETO 2.37**

|  |  |
| --- | --- |
| Nombre del reto: | Detección de Enfermedades Tempranas |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| Un centro de salud desea iniciar una investigación relacionada con la detección temprana de enfermedades asociadas a funciones renales utilizando como indicador el nivel de creatinina en el organismo.  La creatinina se puede medir a través de muestras de laboratorio tanto en sangre como en orina, y sus valores normales son los siguientes:   |  |  |  | | --- | --- | --- | |  | Masculino | Femenino | | Rango normal de creatinina en sangre. Miligramos por decilitro de sangre (mg/dL) | de 0,74 a 1,35 mg/dL | de 0,59 a 1,04 mg/dL | | Rango normal de creatinina en orina. Miligramos por kilogramo de masa corporal por día (mg/kg/día). | De 14 a 26 mg/kg/día | De 11 a 20 mg/kg/día |   El diagnóstico preliminar de una enfermedad crónica renal (ECR) inicia con la toma de 3 muestras del nivel de creatinina de un paciente, las primeras dos muestras corresponden al nivel de creatinina en sangre, y la última al nivel de creatinina en la orina. En cada prueba realizada se le otorga un puntaje de 5 al paciente si el resultado es anormal, y de 0 si no lo es. Al final de las 3 pruebas de acuerdo al puntaje total obtenido se procede con la categorización del riesgo del paciente a padecer una ECR de acuerdo a la siguiente tabla:   |  |  | | --- | --- | | Puntaje | Riesgo | | 0 | Sin riesgo | | 5 | Bajo | | 10 | Medio | | 15 | Alto |   Como apoyo a la fase inicial de la investigación se le ha contratado a usted para realizar un programa que deberá entonces:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, y género (M para masculino y F para femenino) de cada uno de los pacientes. * Leer y almacenar el valor de las muestras de laboratorio de cada uno de los pacientes.   Una vez leídos los datos:   * Por paciente indicar el puntaje total obtenido al evaluar las 3 muestras. * Por paciente indicar la categorización del riesgo. * Por paciente ¿cuántas muestras se encuentran por debajo de los niveles normales?. * Por paciente ¿cuales son los números de muestras (separadas por espacio) que se encuentran por debajo de los niveles normales?. * ¿Cual es el nombre del paciente con el valor más alto en el nivel de creatinina en la orina?   Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente que calcule el puntaje obtenido de acuerdo a los valores de la instancia (no recibir datos por parametro).   Ejemplo:  En la siguiente tabla se muestran los datos para dos pacientes.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Nombre | Cedula | Género | Muestra 1 | Muestra 2 | Muestra 3 | | Pedro Gonzalez | 1098462547 | M | 0.7 | 0.6 | 13 | | Andrea Perez | 72645468 | F | 0.4 | 1 | 10 |   Para el paciente 1:  Puntaje obtenido: 15  Categorización riesgo: Alto  Muestras por debajo de los niveles normales: 3  Número de las muestras que se encuentran por debajo de los niveles normales:1 2 3  Para el paciente 2:  Puntaje obtenido: 10  Categorización riesgo: Medio  Muestras por debajo de los niveles normales: 2  Número de las muestras por debajo de los niveles normales: 1 3  ¿Cual es el nombre del paciente con el valor más alto en el nivel de creatinina en la orina?: Pedro Gonzalez | |

**RETO 2.38**

|  |  |
| --- | --- |
| Nombre del reto: | Detección de Enfermedades Tempranas |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| Un centro de salud desea iniciar una investigación relacionada con la detección temprana de enfermedades asociadas a funciones renales utilizando como indicador el nivel de creatinina en el organismo.  La creatinina se puede medir a través de muestras de laboratorio tanto en sangre como en orina, y sus valores normales son los siguientes:   |  |  |  | | --- | --- | --- | |  | Masculino | Femenino | | Rango normal de creatinina en sangre. Miligramos por decilitro de sangre (mg/dL) | de 0,74 a 1,35 mg/dL | de 0,59 a 1,04 mg/dL | | Rango normal de creatinina en orina. Miligramos por kilogramo de masa corporal por día (mg/kg/día). | De 14 a 26 mg/kg/día | De 11 a 20 mg/kg/día |   El diagnóstico preliminar de una enfermedad crónica renal (ECR) inicia con la toma de 3 muestras del nivel de creatinina de un paciente, las primeras dos muestras corresponden al nivel de creatinina en sangre, y la última al nivel de creatinina en la orina. En cada prueba realizada se le otorga un puntaje de 10 al paciente si el resultado es anormal, y de 0 si no lo es. Al final de las 3 pruebas de acuerdo al puntaje total obtenido se procede con la categorización del riesgo del paciente a padecer una ECR de acuerdo a la siguiente tabla:   |  |  | | --- | --- | | Puntaje | Riesgo | | 0 | Sin riesgo | | 10 | Bajo | | 20 | Medio | | 30 | Alto |   Como apoyo a la fase inicial de la investigación se le ha contratado a usted para realizar un programa que:  El programa deberá entonces:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula y género (M para masculino y F para femenino) de cada uno de los pacientes. * Leer y almacenar el valor de las muestras de laboratorio de cada uno de los pacientes.   Una vez leídos los datos:   * Por paciente indicar el puntaje total obtenido al evaluar las 3 muestras. * Por paciente indicar la categorización del riesgo. * Por paciente indicar el número de la muestra más alta. * ¿Cuál es el nombre del paciente con el valor más alto en cada una de las muestras?   Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente que calcule el puntaje obtenido de acuerdo a los valores de la instancia (no recibir datos por parametro).   Ejemplo:  En la siguiente tabla se muestran los datos para dos pacientes.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Nombre | Cedula | Género | Muestra 1 | Muestra 2 | Muestra 3 | | Julian Arevalo | 1098462547 | M | 0.8 | 0.7 | 15 | | Maria Cordoba | 72645468 | F | 0.5 | 1.3 | 11 |   Para el paciente 1  Puntaje obtenido: 10  Categorización riesgo: Bajo  El número de la muestra las alta es el 3  Para el paciente 2  Puntaje obtenido: 20  Categorización riesgo: Medio  El número de la muestra las alta es el 3  El paciente que tiene la muestra 1 más alta es: Julian Arevalo  El paciente que tiene la muestra 2 más alta es: Maria Cordoba  El paciente que tiene la muestra 3 más alta es: Julian Arevalo | |

**RETO 2.39**

|  |  |
| --- | --- |
| Nombre del reto: | Detección de Enfermedades Tempranas |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| Un centro de salud desea iniciar una investigación relacionada con la detección temprana de enfermedades asociadas a funciones renales utilizando como indicador el nivel de creatinina en el organismo.  La creatinina se puede medir a través de muestras de laboratorio tanto en sangre como en orina, y sus valores normales son los siguientes:   |  |  |  | | --- | --- | --- | |  | Hombres | Mujeres | | Rango normal de creatinina en sangre. Miligramos por decilitro de sangre (mg/dL) | de 0,74 a 1,35 mg/dL | de 0,59 a 1,04 mg/dL | | Rango normal de creatinina en orina. Miligramos por kilogramo de masa corporal por día (mg/kg/día). | De 14 a 26 mg/kg/día | De 11 a 20 mg/kg/día |   El diagnóstico preliminar de una enfermedad crónica renal (ECR) inicia con la toma de 3 muestras del nivel de creatinina de un paciente, las primeras dos muestras corresponden al nivel de creatinina en sangre, y la última al nivel de creatinina en la orina. En cada prueba realizada se le otorga un puntaje de 10 al paciente si el resultado es anormal, y de 0 si no lo es. Al final de las 4 pruebas de acuerdo al puntaje total obtenido se procede con la categorización del riesgo del paciente a padecer una ECR de acuerdo a la siguiente tabla:   |  |  | | --- | --- | | Puntaje | Riesgo | | 0 | Sin riesgo | | 10 | Bajo | | 20-30 | Medio | | 40 | Alto |   Como apoyo a la fase inicial de la investigación se le ha contratado a usted para realizar un programa que:  El programa deberá entonces:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, y género (M para masculino y F para femenino) de cada uno de los pacientes. * Leer y almacenar el valor de las muestras de laboratorio de cada uno de los pacientes.   Una vez leídos los datos:   * Por paciente indicar el puntaje total obtenido al evaluar las 4 muestras. * Por paciente indicar la categorización del riesgo. * Por paciente indicar el promedio de las muestras de la creatinina en sangre y en orina. * ¿Cuál es el número de la cédula del paciente con la primera muestra de creatinina más baja?   Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente que calcule el puntaje obtenido de acuerdo a los valores de la instancia (no recibir datos por parametro).   Ejemplo:  En la siguiente tabla se muestran los datos para dos pacientes.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Nombre | Cedula | Género | Muestra 1 | Muestra 2 | Muestra 3 | Muestra 4 | | Pablo Nieto | 1098462547 | M | 0.8 | 0.7 | 15 | 12 | | Maria Cordoba | 72645468 | F | 0.5 | 1.3 | 11 | 13 |   Para el paciente 1  Puntaje obtenido: 20  Categorización riesgo: Medio  Promedio en sangre: 0.75  Promedio en orina: 13.5  Para el paciente 2  Puntaje obtenido: 20  Categorización riesgo: Medio  Promedio en sangre: 0.9  Promedio en orina: 12.0  La cédula del paciente con la primera muestra de creatinina más baja es: 72645468 | |

**RETO 2.40**

|  |  |
| --- | --- |
| Nombre del reto: | Detección de Enfermedades De Transmisión Alimentaria |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| Un centro de salud desea iniciar una investigación relacionada con las enfermedades de transmisión alimentaria.  En el proceso de diagnóstico el profesional de la salud realiza la valoración inicial del paciente en la cual deberá identificar si se padece o no de alguno o algunos de los siguientes síntomas: Náuseas, vómitos, dolor abdominal, diarrea y fiebre.  Una vez valorados los síntomas se verifica con la siguiente tabla cuál es la posible enfermedad de transmisión alimentaria que pudiera tener el paciente:   |  |  |  | | --- | --- | --- | | **Enfermedad** | **Clasificación** | **Síntomas** | | Staphylococcus aureus | Bacteriano | Náuseas, vómitos, dolor abdominal, fiebre, diarrea | | Bacillus cereus | Bacteriano | Náuseas, vómito | | Taenia saginata | Parasitaria | Dolor abdominal, fiebre | | Norovirus | Viral | Náuseas, vómitos, diarrea, fiebre | | Rotavirus | Viral | Vómitos, diarrea |   Para que el paciente sea diagnosticado de manera efectiva deberá presentar todos los síntomas de la enfermedad, ni más ni menos. En caso tal que el paciente no pueda ser diagnosticado con éxito, será valorado con exámenes adicionales que apoyen el proceso de diagnóstico.  Se han tomado los datos de un conjunto de pacientes con el fin de generar estadísticas que den paso a la investigación requerida.  Conforme a lo anterior, se le ha contratado a usted para apoyar el proceso de investigación desarrollando un programa que implemente los siguientes requerimientos:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, y síntomas padecidos de cada uno de los pacientes.   Una vez leídos los datos   * Calcular el diagnóstico de cada paciente., indicando número de cédula (separados por espacio). * ¿Cuál fue el diagnóstico que más se presentó?, si varios diagnósticos cumplen esta condición tomar el primero. * ¿Cuántos pacientes no pudieron ser diagnosticados?   Requerimiento: Los síntomas deben ser leídos en el siguiente orden:  Náuseas, vómitos, dolor abdominal, diarrea y fiebre.  Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente devuelva el diagnostico del paciente dado un vector de tipo String detallando los síntomas del mismo.   Ejemplo  Los siguientes tres pacientes presentaron los síntomas detallados:  Pablo Nieto: cefalea, dolor abdominal, diarrea, náuseas, vómitos, fiebre  Maria Cordoba: anorexia, dolor abdominal, desnutrición.  Jose Perez: fiebre  El diagnostico de Pablo Nieto fue Salmonella sp, y el de Maria Cordoba fue Taenia saginata.  Diagnostico que más se repitio: Salmonella sp  Pacientes sin diagnostico: 1 | |

**RETO 2.41**

|  |  |
| --- | --- |
| Nombre del reto: | Detección de Enfermedades De Transmisión Alimentaria |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| Un centro de salud desea iniciar una investigación relacionada con las enfermedades de transmisión alimentaria.  En el proceso de diagnóstico el profesional de la salud realiza la valoración inicial del paciente en la cual deberá identificar si se padece o no de alguno o algunos de los siguientes síntomas: Náuseas, vómitos, dolor abdominal, diarrea y fiebre.  Una vez valorados los síntomas se verifica con la siguiente tabla cuál es la posible enfermedad de transmisión alimentaria que pudiera tener el paciente:   |  |  |  | | --- | --- | --- | | **Enfermedad** | **Clasificación** | **Síntomas** | | Staphylococcus aureus | Bacteriano | Náuseas, vómitos, dolor abdominal, fiebre, diarrea | | Bacillus cereus | Bacteriano | Náuseas, vómito | | Taenia saginata | Parasitaria | Dolor abdominal, fiebre | | Norovirus | Viral | Náuseas, vómitos, diarrea, fiebre | | Rotavirus | Viral | Vómitos, diarrea |   Para que el paciente sea diagnosticado de manera efectiva deberá presentar todos los síntomas de la enfermedad, ni más ni menos. En caso tal que el paciente no pueda ser diagnosticado con éxito, será valorado con exámenes adicionales que apoyen el proceso de diagnóstico.  Se han tomado los datos de un conjunto de pacientes con el fin de generar estadísticas que den paso a la investigación requerida.  Conforme a lo anterior, se le ha contratado a usted para apoyar el proceso de investigación desarrollando un programa que implemente los siguientes requerimientos:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, y síntomas padecidos de cada uno de los pacientes.   Una vez leídos los datos   * Calcular el diagnóstico de cada paciente., indicando número de cédula (separados por espacio). * ¿Cuál fue el síntoma que más se presentó?, si varios diagnósticos cumplen esta condición tomar el primero. * ¿Cuál fue el síntoma que menos se presentó?, si varios diagnósticos cumplen esta condición tomar el primero. * ¿Cuántos pacientes fueron diagnosticados con éxito?   Requerimiento: Los síntomas deben ser leídos en el siguiente orden:  Náuseas, vómitos, dolor abdominal, diarrea y fiebre.  Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente devuelva el diagnostico del paciente dado un vector de tipo String detallando los síntomas del mismo.   Ejemplo  Los siguientes tres pacientes presentaron los síntomas detallados:  Pablo Nieto: cefalea, dolor abdominal, diarrea, náuseas, vómitos, fiebre  Maria Cordoba: anorexia, dolor abdominal, desnutrición.  Jose Perez: fiebre  El diagnóstico de Pablo Nieto fue Salmonella sp, y el de Maria Córdoba fue Taenia saginata, Jose Perez no tuvo diagnóstico.  Síntoma que más se repitió: fiebre  Síntoma que menos se repitió: náuseas  Pacientes diagnosticados con éxito: 2 | |

**RETO 2.42**

|  |  |
| --- | --- |
| Nombre del reto: | Detección de Enfermedades De Transmisión Alimentaria |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| Un centro de salud desea iniciar una investigación relacionada con las enfermedades de transmisión alimentaria.  En el proceso de diagnóstico el profesional de la salud realiza la valoración inicial del paciente en la cual deberá identificar si se padece o no de alguno o algunos de los siguientes síntomas Náuseas, vómitos, dolor abdominal, diarrea y fiebre.  Una vez valorados los síntomas se verifica con la siguiente tabla cuál es la posible enfermedad de transmisión alimentaria que pudiera tener el paciente:   |  |  |  | | --- | --- | --- | | **Enfermedad** | **Clasificación** | **Síntomas** | | Staphylococcus aureus | Bacteriano | Náuseas, vómitos, dolor abdominal, fiebre, diarrea | | Bacillus cereus | Bacteriano | Náuseas, vómito | | Taenia saginata | Parasitaria | Dolor abdominal, fiebre | | Norovirus | Viral | Náuseas, vómitos, diarrea, fiebre | | Rotavirus | Viral | Vómitos, diarrea |   Para que el paciente sea diagnosticado de manera efectiva deberá presentar todos los síntomas de la enfermedad, ni más ni menos. En caso tal que el paciente no pueda ser diagnosticado con éxito, será valorado con exámenes adicionales que apoyen el proceso de diagnóstico.  Se han tomado los datos de un conjunto de pacientes con el fin de generar estadísticas que den paso a la investigación requerida.  Conforme a lo anterior, se le ha contratado a usted para apoyar el proceso de investigación desarrollando un programa que implemente los siguientes requerimientos:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, y síntomas padecidos de cada uno de los pacientes.   Una vez leídos los datos   * Calcular el diagnóstico de cada paciente, indicando número de cédula (separados por espacio). * De la clasificación bacteriana ¿Cuál fue el diagnóstico que más se presentó?, si varios diagnósticos cumplen esta condición tomar el primero. * De la clasificación viral ¿Cuál fue el diagnóstico que más se presentó?, si varios diagnósticos cumplen esta condición tomar el primero.   Requerimiento: Los síntomas deben ser leídos en el siguiente orden:  Náuseas, vómitos, dolor abdominal, diarrea y fiebre.  Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente devuelva el diagnostico del paciente dado un vector de tipo String detallando los síntomas del mismo.   Ejemplo  Los siguientes tres pacientes presentaron los síntomas detallados:  Pablo Nieto: cefalea, dolor abdominal, diarrea, náuseas, vómitos, fiebre  Maria Cordoba: anorexia, dolor abdominal, desnutrición.  Jose Perez: fiebre  El diagnóstico de Pablo Nieto fue Salmonella sp, y el de Maria Córdoba fue Taenia saginata, Jose Perez no tuvo diagnóstico.  De la clasificación bacteriana el diagnostico que más se presento fue: Salmonella s  De la clasificación viral el diagnostico que más se presento fue: Norovirus | |

**RETO 2.43**

|  |  |
| --- | --- |
| Nombre del reto: | Estudio De Prevención De Mortalidad |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| En Colombia se desea iniciar una investigación relacionada con las causas de mortalidad en la población.  En estudios previos se ha detectado que las siguientes enfermedades causan gran parte de la mortalidad en los pacientes: cáncer, cardiovasculares, respiratorias, cerebrovasculares, hipertensión y diabetes.  El objetivo de la investigación es realizar un estudio estadístico de las enfermedades anteriormente mencionadas con el fin de mitigar la mortalidad en la población y tomar decisiones en cuanto a los recursos disponibles para atacar cada una de ellas.  Para lo anterior, se tienen los siguientes datos de un conjunto de pacientes:   * Nombre completo * Número de cédula * Edad * Ciudad * EPS * Enfermedad diagnosticada   Conforme a lo mencionado, se le ha contratado a usted para apoyar el proceso de investigación desarrollando un programa que implemente los siguientes requerimientos:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, edad, ciudad, EPS, y enfermedad diagnosticada de cada uno de los pacientes.   Una vez leídos los datos   * ¿Cuál es la edad promedio de los pacientes del estudio? * ¿Cuáles pacientes se encuentran por encima de la edad promedio?, indicar nombre y cédula separados por espacios. * Totalice la cantidad de pacientes por enfermedad.   Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente devuelva la i stancia del paciente con mayor edad.   Ejemplo:  Se tienen los datos de los siguientes pacientes:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Julian | 723456 | 45 | Barranquilla | Sura | Cardiovasculares | | Andres | 734673 | 76 | Barranquilla | Sanitas | Cáncer |   Edad promedio: 60,5  PAcientes por encima del promedio:  Andres 734673 | |

**RETO 2.44**

|  |  |
| --- | --- |
| Nombre del reto: | Estudio De Prevención De Mortalidad |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| En Colombia se desea iniciar una investigación relacionada con las causas de mortalidad en la población.  En estudios previos se ha detectado que las siguientes enfermedades causan gran parte de la mortalidad en los pacientes: cáncer, cardiovasculares, respiratorias, cerebrovasculares, hipertensión y diabetes.  El objetivo de la investigación es realizar un estudio estadístico de las enfermedades anteriormente mencionadas con el fin de mitigar la mortalidad en la población y tomar decisiones en cuanto a los recursos disponibles para atacar cada una de ellas.  Para lo anterior, se tienen los siguientes datos de un conjunto de pacientes:   * Nombre completo * Número de cédula * Edad * Ciudad * EPS * Enfermedad diagnosticada   Conforme a lo mencionado, se le ha contratado a usted para apoyar el proceso de investigación desarrollando un programa que implemente los siguientes requerimientos:   * Leer cantidad y nombre de las diferentes EPS en el estudio. * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, edad, ciudad, EPS, y enfermedad diagnosticada de cada uno de los pacientes.   Una vez leídos los datos:   * ¿Cuál es la enfermedad que más se presenta?, si varios diagnósticos cumplen esta condición tomar el primero. * ¿Cuál es la enfermedad que menos se presenta?, si varios diagnósticos cumplen esta condición tomar el primero. * ¿Cuál es la EPS que registra mayor número de pacientes en el estudio?, si varios diagnósticos cumplen esta condición tomar el primero.   Para el diseño del programa usted debera:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente que dado una instancia de otro Paciente devuelva true si pertenece a la misma EPS, false en caso contrario.   Ejemplo:  Se tienen los datos de los siguientes pacientes:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Julian | 723456 | 45 | Barranquilla | Sura | cardiovasculares | | Andres | 734673 | 76 | Barranquilla | Sanitas | Cancer | | Eva | 104312456 | 32 | Bogota | Sura | Cancer |   Enfermedad que más se presenta: Cancer  Enfermedad que menos se presenta:cardiovasculares  EPS con mayor registro: Sura | |

**RETO 2.45**

|  |  |
| --- | --- |
| Nombre del reto: | Estudio De Prevención De Mortalidad |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| En Colombia se desea iniciar una investigación relacionada con las causas de mortalidad en la población.  En estudios previos se ha detectado que las siguientes enfermedades causan gran parte de la mortalidad en los pacientes: cáncer, cardiovasculares, respiratorias, cerebrovasculares, hipertensión y diabetes.  El objetivo de la investigación es realizar un estudio estadístico de las enfermedades anteriormente mencionadas con el fin de mitigar la mortalidad en la población y tomar decisiones en cuanto a los recursos disponibles para atacar cada una de ellas.  Para lo anterior, se tienen los siguientes datos de un conjunto de pacientes:   * Nombre completo * Número de cédula * Edad * Ciudad * EPS * Enfermedad diagnosticada   Conforme a lo mencionado, se le ha contratado a usted para apoyar el proceso de investigación desarrollando un programa que implemente los siguientes requerimientos:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, edad, ciudad, EPS, y enfermedad diagnosticada de cada uno de los pacientes. * ¿Cuál es la ciudad en la cual se encuentra el mayor número de pacientes? * En la ciudad anterior ¿Cuál es la enfermedad que más se presenta y la que menos se presenta? | |

**RETO 2.46**

|  |  |
| --- | --- |
| Nombre del reto: | Estudio De Prevención De Mortalidad |
| Autor reto: | Karen Ávila Hernández |
| Descripción del reto con su respectiva solución: | |
| En Colombia se desea iniciar una investigación relacionada con las causas de mortalidad en la población.  En estudios previos se ha detectado que las siguientes enfermedades causan gran parte de la mortalidad en los pacientes: cáncer, cardiovasculares, respiratorias, cerebrovasculares, hipertensión y diabetes.  El objetivo de la investigación es realizar un estudio estadístico de las enfermedades anteriormente mencionadas con el fin de mitigar la mortalidad en la población y tomar decisiones en cuanto a los recursos disponibles para atacar cada una de ellas.  Para lo anterior, se tienen los siguientes datos de un conjunto de pacientes:   * Nombre completo * Número de cédula * Edad * Ciudad * EPS * Enfermedad diagnosticada   Conforme a lo mencionado, se le ha contratado a usted para apoyar el proceso de investigación desarrollando un programa que implemente los siguientes requerimientos:   * Leer la cantidad de pacientes del estudio. * Leer y almacenar el nombre, número de cédula, edad, ciudad, EPS, y enfermedad diagnosticada de cada uno de los pacientes.   Una vez leidos los datos:   * ¿Cuántos pacientes hay en cada ciudad? * ¿Cuál es la ciudad en la cual se encuentra el menor número de pacientes?   Para el diseño del programa usted deberá:   * Implementar POO creando una clase llamada Paciente. * Implementar un método dentro de clase Paciente que dado una instancia de otro Paciente devuelva true si pertenece a la misma Ciudad, false en caso contrario   Ejemplo:  Se tienen los datos de los siguientes pacientes:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Julian | 723456 | 45 | Barranquilla | Sura | cardiovasculares | | Andres | 734673 | 76 | Barranquilla | Sanitas | Cancer | | Eva | 104312456 | 32 | Bogota | Sura | Cancer |   Pacientes en Barranquilla: 2  Pacientes en Bogota: 1  Menor numero de pacientes en: Bogota | |

**RETO 2.47**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar cuál fue el nivel de riesgo de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo MEDIO o inferior. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo MEDIO, en caso de no haber ninguno devolver NA. * Indicar el nombre del cuerpo de agua con la clasificación IRCA más baja encontrada y su número identificador.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:     |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nivel de riesgo: MEDIO  Cuerpo de agua 2:  Nivel de riesgo: ALTO  Número de cuerpos de agua con nivel de riesgo MEDIO o inferior: 1  Nombres de los cuerpos de agua que tienen un nivel de riesgo MEDIO: Cienaga de Mallorquin  Nombres del cuerpo de agua con la clasificación IRCA más baja encontrada y su número identificador: Cienaga de Mallorquin 0  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | MEDIO  ALTO  1.00  Cienaga de Mallorquin  Cienaga de Mallorquin 0.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.48**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo BAJO o inferior. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar el nombre del cuerpo de agua con la clasificación IRCA más alta encontrada y su número identificador.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nombre: Cienaga de Mallorquin  Cuerpo de agua 2:  Nombre: Rio Magdalena  Número de cuerpos de agua con nivel de riesgo BAJO o inferior: 0  Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO: NA  Nombre del cuerpo de agua con la clasificación IRCA más alta encontrada y su número identificador: Rio Magdalena 1  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | Cienaga de Mallorquin  Rio Magdalena  0.00  NA  Rio Magdalena 1.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.49**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo SIN RIESGO. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Id\_cuerpo\_agua: 0  Cuerpo de agua 2:  Id\_cuerpo\_agua: 1  Número de cuerpos de agua con nivel de riesgo SIN RIESGO: 0  Nombres de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: NA  Clasificación IRCA promedio de todos los cuerpos de agua ingresados: 37.65  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | 0.00  1.00  0.00  NA  37.65 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.50**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar la clasificación IRCA de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo MEDIO o inferior. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo MEDIO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  IRCA: 30  Cuerpo de agua 2:  IRCA: 45.3  Número de cuerpos de agua con nivel de riesgo MEDIO o inferior: 1  Nombres de los cuerpos de agua que tienen un nivel de riesgo MEDIO: Cienaga de Mallorquin  Clasificación IRCA promedio de todos los cuerpos de agua ingresados: 37.65  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | 30.00  45.30  1.00  Cienaga de Mallorquin  37.65 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.51**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar la clasificación IRCA de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo SIN RIESGO. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más alta encontrada e identificación del cuerpo de agua correspondiente.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  IRCA: 30  Cuerpo de agua 2:  IRCA: 45.3  Número de cuerpos de agua con nivel de riesgo BAJO o inferior: 0  Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO: NA  Clasificación IRCA más alta e identificación del cuerpo de agua correspondiente: 45.3 1  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | 30.00  45.30  0.00  NA  45.30 1.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.52**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar la clasificación IRCA de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre SIN RIESGO y BAJO. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  IRCA: 30  Cuerpo de agua 2:  IRCA: 45.3  Número de cuerpos de agua que tienen un nivel de riesgo entre SIN RIESGO y BAJO: 0  Nombres de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: NA  Clasificación IRCA promedio de todos los cuerpos de agua ingresados: 37.65  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | 30.00  45.30  0.00  NA  37.65 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.53**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre BAJO y MEDIO. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nombre: Cienaga de Mallorquin  Cuerpo de agua 2:  Nombre: Rio Magdalena  Número de cuerpos de agua que tienen un nivel de riesgo entre BAJO y MEDIO: 1  Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO: NA  Clasificación IRCA promedio de todos los cuerpos de agua ingresados: 37.65  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | Cienaga de Mallorquin  Rio Magdalena  1.00  NA  37.65 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.54**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre BAJO y MEDIO. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo MEDIO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más alta encontrada entre todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Id\_cuerpo\_agua: 0  Cuerpo de agua 2:  Id\_cuerpo\_agua: 1  Número de cuerpos de agua que tienen un nivel de riesgo entre BAJO y MEDIO: 1  Nombres de los cuerpos de agua que tienen un nivel de riesgo MEDIO: Cienaga de Mallorquin  Clasificación IRCA más alta encontrada entre todos los cuerpos de agua ingresados: 45.30  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | 0.00  1.00  1.00  Cienaga de Mallorquin  45.30 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.55**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre SIN RIESGO y MEDIO. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más baja encontrada entre todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Id\_cuerpo\_agua: 0  Cuerpo de agua 2:  Id\_cuerpo\_agua: 1  Número de cuerpos de agua que tienen un nivel de riesgo entre BAJO y MEDIO: 1  Nombres de los cuerpos de agua que tienen un nivel de riesgo MEDIO: Cienaga de Mallorquin  Clasificación IRCA más baja encontrada entre todos los cuerpos de agua ingresados: 30.00  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | 0.00  1.00  1.00  Cienaga de Mallorquin  30.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.56**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre MEDIO y ALTO. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo ALTO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nombre: Cienaga de Mallorquin  Cuerpo de agua 2:  Nombre: Rio Magdalena  Número de cuerpos de agua que tienen un nivel de riesgo entre MEDIO y ALTO: 2  Nombres de los cuerpos de agua que tienen un nivel de riesgo ALTO: Rio Magdalena  Clasificación IRCA promedio de todos los cuerpos de agua ingresados: 37.65  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | Cienaga de Mallorquin  Rio Magdalena  2.00  Rio Magdalena  37.65 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.57**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre MEDIO y ALTO. * Indicar el nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo MEDIO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Id\_cuerpo\_agua: 0  Cuerpo de agua 2:  Id\_cuerpo\_agua: 1  Número de cuerpos de agua que tienen un nivel de riesgo entre MEDIO y ALTO: 2  Nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo MEDIO: Barranquilla  Clasificación IRCA promedio de todos los cuerpos de agua ingresados: 37.65  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | 0  1  2.00  Barranquilla  37.65 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.58**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectivo municipio. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre ALTO e INVIABLE SANITARIAMENTE. * Indicar el nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo ALTO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo más alto entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nombre de cuerpo de agua y su respectivo municipio: Cienaga de Mallorquin Barranquilla  Cuerpo de agua 2:  Nombre de cuerpo de agua y su respectivo municipio: Rio Magdalena Barranquilla  Número de cuerpos de agua que tienen un nivel de riesgo entre ALTO e INVIABLE SANITARIAMENTE: 1  Nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo ALTO: Barranquilla  Nivel de riesgo más alto entre de todos los cuerpos de agua ingresados: ALTO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | Cienaga de Mallorquin Barranquilla  Rio Magdalena Barranquilla  1.00  Barranquilla  ALTO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.59**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectivo indicador. * Indicar cuántos cuerpos de agua tienen un nivel de riesgo entre ALTO e INVIABLE SANITARIAMENTE. * Indicar el nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo MEDIO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo más bajo entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nombre de cuerpo de agua y su respectivo indicador: Cienaga de Mallorquin 0  Cuerpo de agua 2:  Nombre de cuerpo de agua y su respectivo indicador: Rio Magdalena 1  Número de cuerpos de agua que tienen un nivel de riesgo entre ALTO e INVIABLE SANITARIAMENTE: 1  Nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo ALTO: Barranquilla  Nivel de riesgo más bajo entre de todos los cuerpos de agua ingresados: MEDIO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | Cienaga de Mallorquin 0  Rio Magdalena 1  1.00  Barranquilla  MEDIO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.60**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectiva clasificación IRCA. * Indicar cuántos cuerpos de agua tienen una clasificación IRCA mayor o igual a 50. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo ALTO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más alta entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nombre de cuerpo de agua y su respectiva clasificación IRCA: Cienaga de Mallorquin 30  Cuerpo de agua 2:  Nombre de cuerpo de agua y su respectiva clasificación IRCA: Rio Magdalena 45.3  Número de cuerpos de agua que tienen una clasificación IRCA mayor o igual a 50: 0  Nombres de los cuerpos de agua que tienen un nivel de riesgo ALTO: Rio Magdalena  Clasificación IRCA más alta entre de todos los cuerpos de agua ingresados: 45.3  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | Cienaga de Mallorquin 30.00  Rio Magdalena 45.30  0.00  Rio Magdalena  45.30 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.61**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua y su respectiva clasificación IRCA. * Indicar cuántos cuerpos de agua tienen una clasificación IRCA menor o igual a 50. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más baja entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | Cienaga de Mallorquin | 0 | Barranquilla | 30 | | Rio Magdalena | 1 | Barranquilla | 45.3 |   Cuerpo de agua 1:  Nombre de cuerpo de agua y su respectiva clasificación IRCA: Cienaga de Mallorquin 30  Cuerpo de agua 2:  Nombre de cuerpo de agua y su respectiva clasificación IRCA: Rio Magdalena 45.3  Número de cuerpos de agua que tienen una clasificación IRCA menor o igual a 50: 2  Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO: NA  Clasificación IRCA más baja entre de todos los cuerpos de agua ingresados: 30.00  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “Cienaga de Mallorquin” 0 “Barranquilla” 30  “Rio Magdalena” 1 “Barranquilla” 45.3 | Cienaga de Mallorquin 30.00  Rio Magdalena 45.30  2.00  NA  30.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.62**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua y su respectivo nivel de riesgo. * Indicar cuántos cuerpos de agua tienen una clasificación IRCA menor o igual a 14. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo inferior a MEDIO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más alta entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Identificador de cada cuerpo de agua y su respectivo nivel de riesgo: 2 BAJO  Cuerpo de agua 2:  Identificador de cada cuerpo de agua y su respectivo nivel de riesgo: 3 SIN RIESGO  Número de cuerpos de agua que tienen una clasificación IRCA menor o igual a 14: 2  Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO: La cuenca del Mar Caribe  Clasificación IRCA más alta entre de todos los cuerpos de agua ingresados: 14  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | 2.00 BAJO  3.00 SIN RIESGO  2.00  La cuenca del Mar Caribe  14 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.63**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectivo nivel de riesgo. * Indicar cuántos cuerpos de agua tienen una clasificación IRCA mayor a 14. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo inferior a MEDIO y mayor que SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nombre de cada cuerpo de agua y su respectivo nivel de riesgo: La cuenca del Mar Caribe BAJO  Cuerpo de agua 2:  Nombre de cada cuerpo de agua y su respectivo nivel de riesgo: Embalse del Guajaro SIN RIESGO  Número de cuerpos de agua que tienen una clasificación IRCA mayor a 14: 0  Nombres de los cuerpos de agua que tienen un nivel de riesgo inferior a MEDIO y mayor que SIN RIESGO: NA  Clasificación IRCA promedio entre de todos los cuerpos de agua ingresados: 9  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | La cuenca del Mar Caribe BAJO  Embalse del Guajaro SIN RIESGO  0.00  NA  9.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.64**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectivo identificador. * Indicar cuántos cuerpos de agua tienen una clasificación IRCA mayor que 0. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo entre BAJO y ALTO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más alta y la más baja entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nombre de cada cuerpo de agua y su respectivo identificador: La cuenca del Mar Caribe 2  Cuerpo de agua 2:  Nombre de cada cuerpo de agua y su respectivo identificador: Embalse del Guajaro 3  Número de cuerpos de agua que tienen una clasificación IRCA mayor a 0: 2  Nombres de los cuerpos de agua que tienen un nivel de riesgo entre BAJO y ALTO: La cuenca del Mar Caribe  Clasificación IRCA más alta y la más baja entre de todos los cuerpos de agua ingresados: 14 4  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | La cuenca del Mar Caribe 2.00  Embalse del Guajaro 3.00  2.00  La cuenca del Mar Caribe  14.00 4.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.65**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nivel de riesgo de cada cuerpo de agua y su respectivo identificador. * Indicar cuántos cuerpos de agua tienen una clasificación IRCA menor que 50. * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo entre SIN RIESGO y BAJO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más alta y la más baja entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nivel de riesgo de cada cuerpo de agua y su respectivo identificador: BAJO 2  Cuerpo de agua 2:  Nivel de riesgo de cada cuerpo de agua y su respectivo identificador: SIN RIESGO 3  Número de cuerpos de agua que tienen una clasificación IRCA menor que 50: 2  Nombres de los cuerpos de agua que tienen un nivel de riesgo entre SIN RIESGO y BAJO: La cuenca del Mar Caribe Embalse del Guajaro  Clasificación IRCA más alta y la más baja entre de todos los cuerpos de agua ingresados: 14 4  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | BAJO 2.00  SIN RIESGO 3.00  2.00  La cuenca del Mar Caribe Embalse del Guajaro  14.00 4.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.66**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nivel de riesgo de cada cuerpo de agua y su respectiva calificación IRCA. * Indicar cuántos cuerpos de agua tienen una clasificación IRCA mayor que 50. * Indicar el nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nivel de riesgo de cada cuerpo de agua y su respectiva calificación IRCA: BAJO 14  Cuerpo de agua 2:  Nivel de riesgo de cada cuerpo de agua y su respectiva calificación IRCA: SIN RIESGO 4  Número de cuerpos de agua que tienen una clasificación IRCA menor que 50: 0  Nombre de los municipios de los cuerpos de agua que tienen un nivel de riesgo entre BAJO: La cuenca del Mar Caribe  Clasificación IRCA promedio entre de todos los cuerpos de agua ingresados: 9  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | BAJO 14.00  SIN RIESGO 4.00  0.00  La cuenca del Mar Caribe  9.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.67**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nivel de riesgo de cada cuerpo de agua y su respectiva calificación IRCA. * Indicar cuántos cuerpos de agua requieren la acción de la GOBERNACION. * Indicar el identificador de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nivel de riesgo de cada cuerpo de agua y su respectiva calificación IRCA: BAJO 14  Cuerpo de agua 2:  Nivel de riesgo de cada cuerpo de agua y su respectiva calificación IRCA: SIN RIESGO 4  Número de cuerpos de agua que requieren la acción de la GOBERNACION: 0  Identificador de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: 3  Clasificación IRCA promedio entre de todos los cuerpos de agua ingresados: 9  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | BAJO 14.00  SIN RIESGO 4.00  0.00  3.00  9.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.68**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nivel de riesgo de cada cuerpo de agua y su respectivo identificador. * Indicar cuántos cuerpos de agua no requieren la acción de la GOBERNACION. * Indicar la calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nivel de riesgo de cada cuerpo de agua y su respectivo identificador: BAJO 2  Cuerpo de agua 2:  Nivel de riesgo de cada cuerpo de agua y su respectivo identificador: SIN RIESGO 3  Número de cuerpos de agua que no requieren la acción de la GOBERNACION: 2  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: 4  Clasificación IRCA promedio entre de todos los cuerpos de agua ingresados: 9  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | BAJO 2.00  SIN RIESGO 3.00  2.00  4.00  9.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.69**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nivel de riesgo de cada cuerpo de agua y su respectivo identificador. * Indicar cuántos cuerpos de agua requieren la acción de la ALCALDIA. * Indicar la calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nivel de riesgo de cada cuerpo de agua y su respectivo identificador: BAJO 2  Cuerpo de agua 2:  Nivel de riesgo de cada cuerpo de agua y su respectivo identificador: SIN RIESGO 3  Número de cuerpos de agua que no requieren la acción de la ALCALDIA: 0  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo BAJO: 14  Clasificación IRCA promedio entre de todos los cuerpos de agua ingresados: 9  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | BAJO 2.00  SIN RIESGO 3.00  0.00  14.00  9.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.70**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nivel de riesgo de cada cuerpo de agua y su respectivo nombre. * Indicar cuántos cuerpos de agua no requieren la acción de la ALCALDIA. * Indicar la calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nivel de riesgo de cada cuerpo de agua y su respectivo nombre: BAJO La cuenca del Mar Caribe  Cuerpo de agua 2:  Nivel de riesgo de cada cuerpo de agua y su respectivo nombre: SIN RIESGO Embalse del Guajaro  Número de cuerpos de agua que no requieren la acción de la ALCALDIA: 2  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: 4  Clasificación IRCA promedio entre de todos los cuerpos de agua ingresados: 9  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | BAJO La cuenca del Mar Caribe  SIN RIESGO Embalse del Guajaro  2.00  4.00  9.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.71**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nivel de riesgo de cada cuerpo de agua y su respectivo identificador. * Indicar cuántos cuerpos de agua requieren la acción de la ALCALDIA o de la GOBERNACION. * Indicar la calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo MEDIO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más baja entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nivel de riesgo de cada cuerpo de agua y su respectivo nombre: BAJO 2  Cuerpo de agua 2:  Nivel de riesgo de cada cuerpo de agua y su respectivo nombre: SIN RIESGO 3  Número de cuerpos de agua que requieren la acción de la ALCALDIA o de la GOBERNACION: 0  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo MEDIO: 0  Clasificación IRCA más baja entre de todos los cuerpos de agua ingresados: 4  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | BAJO 2.00  SIN RIESGO 3.00  0.00  0.00  4.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.72**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectivo identificador. * Indicar cuántos cuerpos de agua requieren la acción de la PERSONA PRESTADORA. * Indicar la calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar la clasificación IRCA más alta entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nombre y su respectivo identificador: La cuenca del Mar Caribe 2  Cuerpo de agua 2:  Nombre y su respectivo identificador: Embalse del Guajaro 3  Número de cuerpos de agua que requieren la acción de la PERSONA PRESTADORA: 1  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo BAJO: 14  Clasificación IRCA más alta entre de todos los cuerpos de agua ingresados: 14  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | La cuenca del Mar Caribe 2.00  Embalse del Guajaro 3.00  0.00  14.00  14.00 |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.73**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectivo identificador. * Indicar cuántos cuerpos de agua no requieren la acción de la PERSONA PRESTADORA. * Indicar la calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo ALTO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo más alto entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nombre y su respectivo identificador: La cuenca del Mar Caribe 2  Cuerpo de agua 2:  Nombre y su respectivo identificador: Embalse del Guajaro 3  Número de cuerpos de agua que no requieren la acción de la PERSONA PRESTADORA: 1  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo ALTO: NA  Nivel de riesgo más alto entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | La cuenca del Mar Caribe 2.00  Embalse del Guajaro 3.00  1.00  NA  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.74**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectiva calificación IRCA. * Indicar cuántos cuerpos de agua requieren la acción de la PERSONA PRESTADORA o la ALCALDIA. * Indicar el identificador de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo más bajo entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nombre y su respectiva calificación IRCA: La cuenca del Mar Caribe 14  Cuerpo de agua 2:  Nombre y su respectiva calificación IRCA: Embalse del Guajaro 4  Número de cuerpos de agua que requieren la acción de la PERSONA PRESTADORA o la ALCALDIA: 1  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: 4  Nivel de riesgo más bajo entre de todos los cuerpos de agua ingresados: SIN RIESGO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | La cuenca del Mar Caribe 14.00  Embalse del Guajaro 4.00  1.00  4.00  SIN RIESGO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.75**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectiva entidad a tomar acción. * Indicar cuántos cuerpos de agua no requieren la acción de la PERSONA PRESTADORA o la ALCALDIA. * Indicar la calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo más alto entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nombre y su respectiva entidad a tomar acción: La cuenca del Mar Caribe PERSONA PRESTADORA  Cuerpo de agua 2:  Nombre y su respectiva entidad a tomar acción: Embalse del Guajaro CONTINUAR VIGILANCIA  Número de cuerpos de agua que no requieren la acción de la PERSONA PRESTADORA o la ALCALDIA: 1  Calificación IRCA de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: 4  Nivel de riesgo más alto entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | La cuenca del Mar Caribe PERSONA PRESTADORA  Embalse del Guajaro CONTINUAR VIGILANCIA  1.00  4.00  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.76**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua y su respectiva entidad a tomar acción. * Indicar cuántos cuerpos de agua requieren solamente CONTINUAR VIGILANCIA * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Id\_cuerpo\_agua y su respectiva entidad a tomar acción: 2 PERSONA PRESTADORA  Cuerpo de agua 2:  Id\_cuerpo\_agua y su respectiva calificación IRCA: 3 CONTINUAR VIGILANCIA  Número de cuerpos de agua que requieren solamente CONTINUAR VIGILANCIA: 1  Nombres de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO: Embalse del Guajaro  Nivel de riesgo promedio entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | 2.00 PERSONA PRESTADORA  3.00 CONTINUAR VIGILANCIA  1.00  Embalse del Guajaro  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.77**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua y su respectiva entidad a tomar acción. * Indicar cuántos cuerpos de agua requieren desafortunadamente más que CONTINUAR VIGILANCIA * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo ALTO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Id\_cuerpo\_agua y su respectiva entidad a tomar acción: 2 PERSONA PRESTADORA  Cuerpo de agua 2:  Id\_cuerpo\_agua y su respectiva calificación IRCA: 3 CONTINUAR VIGILANCIA  Número de cuerpos de agua que requieren más que CONTINUAR VIGILANCIA: 1  Nombres de los cuerpos de agua que tienen un nivel de riesgo ALTO: NA  Nivel de riesgo promedio entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | 2.00 PERSONA PRESTADORA  3.00 CONTINUAR VIGILANCIA  1.00  NA  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.78**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el nombre de cada cuerpo de agua y su respectiva entidad a tomar acción. * Indicar cuántos cuerpos de agua requieren más que la acción de la PERSONA PRESTADORA * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo INVIABLE SANITARIAMENTE, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  Nombre y su respectiva entidad a tomar acción: La cuenca del Mar Caribe PERSONA PRESTADORA  Cuerpo de agua 2:  Nombre y su respectiva calificación IRCA: Embalse del Guajaro CONTINUAR VIGILANCIA  Número de cuerpos de agua que requieren más que la acción de la PERSONA PRESTADORA: 0  Nombres de los cuerpos de agua que tienen nivel de riesgo INVIABLE SANITARIAMENTE: NA  Nivel de riesgo promedio entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | La cuenca del Mar Caribe PERSONA PRESTADORA  Embalse del Guajaro CONTINUAR VIGILANCIA  0.00  NA  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.79**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua y su respectiva entidad a tomar acción. * Indicar cuántos cuerpos de agua NO requieren más que la acción de la PERSONA PRESTADORA * Indicar el Nombres de los cuerpos de agua que tienen un nivel de riesgo BAJO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  identificador y su respectiva entidad a tomar acción: 2 PERSONA PRESTADORA  Cuerpo de agua 2:  identificador y su respectiva calificación IRCA: 3 CONTINUAR VIGILANCIA  Número de cuerpos de agua que no requieren más que la acción de la PERSONA PRESTADORA: 2  Nombres de los cuerpos de agua que tienen nivel de riesgo BAJO: La cuenca del Mar Caribe  Nivel de riesgo promedio entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | 2.00 PERSONA PRESTADORA  3.00 CONTINUAR VIGILANCIA  2.00  La cuenca del Mar Caribe  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.80**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua y su respectiva entidad a tomar acción. * Indicar la suma de las clasificaciones IRCA de los cuerpos de agua ingresados. * Indicar la clasificación IRCA de los cuerpos de agua que tienen un nivel de riesgo SIN RIESGO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  identificador y su respectiva entidad a tomar acción: 2 PERSONA PRESTADORA  Cuerpo de agua 2:  identificador y su respectiva calificación IRCA: 3 CONTINUAR VIGILANCIA  Suma de las clasificaciones IRCA de los cuerpos de agua ingresados: 18  Calificación IRCA de cuerpos de agua que tienen nivel de riesgo SIN RIESGO: 4  Nivel de riesgo promedio entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | 2.00 PERSONA PRESTADORA  3.00 CONTINUAR VIGILANCIA  18.00  4.00  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**RETO 2.81**

|  |  |
| --- | --- |
| Nombre del reto: | Lector del nivel de la calidad del agua en el Atlántico |
| Autora reto: | Camila Lozano Villalba |
| Descripción de reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.  El departamento del Atlántico se ha comprometido con esta causa y por ello ha decidido adoptar estos retos, se lista uno de los principales relacionados con el agua potable:  De aquí a 2030, se busca lograr el acceso universal y equitativo al agua potable a un precio asequible para todos.  Algunas ONG’s se atribuyeron la tarea de poder diseñar un dispositivo para analizar la calidad  del agua de poblaciones apartadas. Para comenzar, requieren que el dispositivo cuente con un  lector de la calidad del agua. Después de la lectura, el dispositivo nos entrega el índice de riesgo  de la calidad del agua, IRCA, y según este resultado debe indicar el nivel de riesgo.   |  |  |  |  | | --- | --- | --- | --- | | **Clasificación IRCA (%)** | **Nivel de riesgo** | **Entidades a notificar** | **Entidades a tomar**  **acciones** | | 80.1 - 100 | INVIABLE  SANITARIAMENTE | Persona prestadora, COVE, Alcaldía, Gobernación, SSPD, MPS, INS, MAVDT, Contraloría General, Procuraduría General | GOBERNACION | | 35.1 - 80 | ALTO | Persona prestadora, COVE,  Alcaldía, Gobernación, SSPD | ALCALDIA | | 14.1 - 35 | MEDIO | Persona prestadora, COVE, Alcaldía, Gobernación | PERSONA PRESTADORA | | 5.1 - 14 | BAJO | Persona prestadora, COVE | | 0 - 5 | SIN RIESGO | Continuar el control y la  vigilancia | CONTINUAR VIGILANCIA |   **Se requiere que usted desarrolle un programa que debe realizar lo siguiente:**   * Leer la cantidad de cuerpos de agua a analizar. * Leer y almacenar el nombre de cada cuerpo de agua, número con el que se identifica, el municipio del que forma parte y la clasificación IRCA.   **Una vez leídos los datos:**   * Indicar el identificador de cada cuerpo de agua y su respectivo nombre. * Indicar la resta de las clasificaciones IRCA de los cuerpos de agua ingresados. * Indicar la clasificación IRCA de los cuerpos de agua que tienen un nivel de riesgo ALTO, en caso de no haber ninguno devolver NA. * Indicar el nivel de riesgo promedio entre de todos los cuerpos de agua ingresados.   **En cuanto al diseño del programa se debe realizar lo siguiente:**  Implementar POO creando una clase llamada CuerpoDeAgua.  Implementar un método dentro de clase CuerpoDeAgua que calcule el nivel de riesgo y la entidad a tomar acción de un cuerpo de agua de acuerdo a los valores de la instancia (no recibir datos por parámetro).  Ejemplo:   |  |  |  |  | | --- | --- | --- | --- | | Nombre | Id\_cuerpo\_agua | Municipio | IRCA | | La cuenca del Mar Caribe | 2 | Puerto Colombia | 14 | | Embalse del Guajaro | 3 | Sabanalarga | 4 |   Cuerpo de agua 1:  identificador y su respectivo nombre: 2 La cuenca del Mar Caribe  Cuerpo de agua 2:  identificador y su respectiva calificación IRCA: 3 Embalse del Guajaro  Resta de las clasificaciones IRCA de los cuerpos de agua ingresados: -18  Calificación IRCA de cuerpos de agua que tienen nivel de riesgo ALTO: NA  Nivel de riesgo promedio entre de todos los cuerpos de agua ingresados: BAJO  **Formato entrada y salida esperadas:**   |  |  | | --- | --- | | Entrada esperada | Salida esperada | | 2  “La cuenca del Mar Caribe” 2 “Puerto Colombia” 14  “Embalse del Guajaro” 3 “Sabanalarga” 4 | 2.00 La cuenca del Mar Caribe  3.00 Embalse del Guajaro  -18.00  NA  BAJO |   **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas  y salidas del programa. Por favor NO use ningún signo dentro del desarrollo de su solución ya  que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** El archivo debe llamarse reto2.java, de lo contrario no podrá ser cargado en la plataforma de Codegrade.  **Nota:** Los resultados numéricos deben estar formateados a dos cifras decimales. | |

**Reto 2.82**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima las alertas que presentaron los pacientes * Imprima los tratamientos que deben seguir los pacientes   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | moderadamente sospechoso  levemente sospechoso  benigno  aaf  seguimiento  no aaf | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | levemente sospechoso  altamente sospechoso  seguimiento  aaf |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.83**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima el conteo de las alertas que presentaron los pacientes. Primero imprimir la alerta y separado por un espacio el conteo. El orden de impresión debe ser: benigno, no sospechoso, levemente sospechoso, moderadamente sospechoso y altamente sospechoso. * Imprima el conteo de los tratamientos que presentaron los pacientes. Primero imprimir la alerta y separado por un espacio el conteo. El orden de impresión debe ser: no aaf, seguimiento y aaf   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | benigno 1  levemente sospechoso 1  moderadamente sospechoso 1  no aaf 1  seguimiento 1  aaf 1 | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | levemente sospechoso 1  altamente sospechoso 1  seguimiento 1  aaf 1 |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.84**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima el promedio del tamaño de los nódulos con formato de strings a dos cifras decimales * Imprima el mínimo de los tamaños de los nódulos   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | 1.67  1.0 | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | 1.50  1.0 |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.85**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima la desviación estándar del tamaño de los nódulos con formato de strings a dos cifras decimales * Imprima el máximo de los tamaños de los nódulos   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | 0.58  2.0 | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | 0.71  2.0 |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.86**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima la varianza del tamaño de los nódulos con formato de strings a dos cifras decimales * Imprima la suma del tamaño de los nódulos   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | 0.33  5.0 | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | 0.50  3.0 |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.87**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima las modas de las alertas de los pacientes:   + Si es una sola moda: imprimirla   + Si son dos, tres o cuatro modas: imprimir las alertas en el siguiente orden en una línea nueva: benigno, no sospechoso, levemente sospechoso, moderadamente sospechoso y altamente sospechoso   + Si todas las alertas tienen la misma frecuencia imprimir: no hay moda * Imprima las modas de los tratamientos:   + Si es una sola moda: imprimirla   + Si son dos: imprimir los tratamientos en el siguiente orden en una línea nueva: no aaf, seguimiento, aaf   + Si todos los tratamientos tienen la misma frecuencia, imprimir: no hay moda   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | no hay moda  no hay moda | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | altamente sospechoso  aaf |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.88**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima la alerta de los pacientes menos frecuente:   + Si es un solo dato: imprimirlo   + Si son dos, tres o cuatro datos: imprimir las alertas en el siguiente orden en una línea nueva: benigno, no sospechoso, levemente sospechoso, moderadamente sospechoso y altamente sospechoso   + Si todas las alertas tienen la misma frecuencia imprimir: todos los datos tienen la misma frecuencia * Imprima los tratamientos de los pacientes menos frecuente:   + Si es un solo dato: imprimirlo   + Si son dos: imprimir los tratamientos en el siguiente orden en una línea nueva: no aaf, seguimiento, aaf   + Si todos los tratamientos tienen la misma frecuencia, imprimir: todos los datos tienen la misma frecuencia   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | todos los datos tienen la misma frecuencia  todos los datos tienen la misma frecuencia | | **Entrada esperada** | | 4  C3 E4 F1 M3 0 1 1 0 1.63  C4 E2 F1 M1 0 0 0 1 0.54  C4 E3 F1 M3 1 1 0 0 2.27  C2 E3 F1 M3 0 1 0 1 2.73 | | **Salida esperada** | | moderadamente sospechoso  seguimiento |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.89**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro * Imprima el rango del tamaño de los nódulos de los pacientes * Imprima la mediana del tamaño de los nódulos de los pacientes   **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | 1.0  2.0 | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | 1.0  **1.5** |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.90**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Imprima las alertas de los nódulos de los pacientes, en el orden en que aparecen en la tabla, en una línea nueva, y separada por un espacio la proporción con formato de strings a dos cifras decimales * Imprima los tratamientos de los nódulos de los pacientes, en el orden en que aparecen en la tabla, en una línea nueva, y separada por un espacio la proporción con formato de strings a dos cifras decimales * Calcular la media geométrica de las alertas utilizando las proporciones * Calcular la media geométrica de los tratamientos utilizando las proporciones * Imprimir la media geométrica de las alertas con formato de strings a dos cifras decimales * Imprimir la media geométrica de los tratamientos con formato de strings a dos cifras decimales   Utilice la siguiente fórmula para el cálculo de la media geométrica:  **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | benigno 0.33  levemente sospechoso 0.33  moderadamente sospechoso 0.33  no aaf 0.33  seguimiento 0.33  aaf 0.33  0.33  0.33 | | **Entrada esperada** | | 4  C3 E4 F1 M3 0 1 1 0 1.63  C4 E2 F1 M1 0 0 0 1 0.54  C4 E3 F1 M3 1 1 0 0 2.27  C2 E3 F1 M3 0 1 0 1 2.73 | | **Salida esperada** | | moderadamente sospechoso 0.25  altamente sospechoso 0.75  seguimiento 0.25  aaf 0.75  0.43  0.43 |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |

**Reto 2.91**

|  |  |
| --- | --- |
| Nombre del reto: | Clasificación de nódulos tiroideos y acciones a tomar |
| Autor reto: | Jhon Jaime de Jesús Corro Pareja |
| Descripción del reto con su respectiva solución: | |
| En el año 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. La buena salud es esencial para el desarrollo sostenible, y la Agenda 2030 busca reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de la alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.  El TI-RADS (Thyroid Imaging Reporting and Data Systems) es una prueba para la clasificación de nódulos tiroideos. Basándose en cinco características de los nodos como lo son la composición, ecogenicidad, forma márgenes y focos ecogénicos. Cada característica tiene un puntaje y según el total de punto se describe si el nódulo es benigno o no y, además, si requiere seguimiento o intervención con aguja fina.  El Ministerio de Salud le pide que lo ayude en el desarrollo de un programa para la emisión de alertas tempranas de acuerdo a las características del nodo. Para ello debe tener en cuenta las tablas siguientes:   |  |  |  | | --- | --- | --- | | **Composición** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | C1 | Puntos quísticos o casi completamente quísticos | 0 | | C2 | Espongiforme | 0 | | C3 | Mixto quístico y sólido | 1 | | C4 | Sólido o casi completamente sólido | 2 |  |  |  |  | | --- | --- | --- | | **Ecogenicidad** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | E1 | Anecoico | 0 | | E2 | Hiperecoico o isoecoico | 1 | | E3 | Hipoecoico | 2 | | E4 | Muy hipoecoico | 3 |  |  |  |  | | --- | --- | --- | | **Forma** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | F1 | Más ancho que alto | 0 | | F2 | Más alto que ancho | 3 |  |  |  |  | | --- | --- | --- | | **Margen** | | | | **SE DEBE ESCOGER SOLO UNO** | | | | **Código** | **Clasificación** | **Puntaje** | | M1 | Suave | 0 | | M2 | Bien definido | 0 | | M3 | Lobulado o irregular | 2 | | M4 | Extensión extra-tiroidea | 3 |  |  |  |  | | --- | --- | --- | | **Focos ecogénicos** | | | | **SE DEBEN SELECCIONAR TODOS LOS QUE APLIQUEN** | | | | **Código** | **Clasificación** | **Puntaje** | | FE1 | Ninguno o grandes artefactos en cola de cometa | 0 | | FE2 | Macrocalcificaciones | 1 | | FE3 | Calcificaciones periféricas (borde) | 2 | | FE4 | Focos ecogénicos punteados | 3 |   Después de calculado el puntaje la clasificación de los nódulos es la siguiente:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 - 1 puntos | 2 puntos | 3 puntos | 4 - 6 puntos | 7 o más | | Benigno | No sospechoso | Levemente sospechoso | Moderadamente sospechoso | Altamente sospechoso | | No AAF | No AAF | AAF si tamaño ≥ 2.5  Seguimiento si tamaño < 2.5 | AAF si tamaño ≥ 1.5  Seguimiento si tamaño < 1.5 | AAF si tamaño ≥ 1  Seguimiento si tamaño < 1 |   \*\* AAF = aspiración con aguja final.  Este desarrollo solicita que el programa utilice programación orientada a objetos y por lo tanto el programa deberá implementar una clase llamada Nodule que reciba como atributos las características de los nódulos tiroideos que se especifican en las tablas. Además, se solicita que el programa:   * Lea una variable N que indique el número de pacientes que se analizarán * Lea los datos de los N pacientes e instancie la clase Nodule con estos * Implemente el método dentro de la clase para calcular la alerta de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Implemente el método dentro de la clase para calcular el tratamiento de acuerdo con los valores de la instancia (no recibir datos por parámetro) * Calcule un vector donde los elementos sean los tamaños de los nódulos divididos entre el valor máximo * Imprima las alertas de los nódulos de los pacientes * Imprima los tratamientos de los nódulos de los pacientes * Imprima cada elementos del vector calculado con formato de strings a dos cifras decimales   Utilice la siguiente fórmula para el cálculo de la media geométrica:  **Ejemplos:**   |  | | --- | | **Entrada esperada** | | 3  C1 E1 F2 M3 0 1 0 0 2  C1 E1 F1 M1 0 1 1 0 1  C1 E1 F1 M1 1 0 0 0 2 | | **Salida esperada** | | moderadamente sospechoso  levemente sospechoso  benigno  aaf  seguimiento  no aaf  1.00  0.50  1.00 | | **Entrada esperada** | | 2  C1 E1 F1 M1 0 1 1 0 1  C4 E4 F2 M4 1 1 1 1 2 | | **Salida esperada** | | levemente sospechoso  altamente sospechoso  seguimiento  aaf  0.50  1.00 |   **Nota:** Tenga que cuenta que múltiples valores de entrada pueden estar en una misma línea separados por espacio.  **Nota:** Por favor **NO** incluya mensajes en los inputs.  **Nota:** Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. **Por favor NO use ningún signo ortográfico dentro del desarrollo de su solución** ya que estos pueden representar errores en la calificación automática de Codegrade.  **Nota:** El archivo debe llamarse **reto2.java**, de lo contrario no podrá ser cargado en la plataforma de Codegrade. | |