

Indicaciones específicas:

- Esta evaluación contiene 8 páginas (incluyendo esta página) con 1 pregunta. El total de puntos es 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un archivo diferente. Se deberá usar los siguientes nombres de archivos:
 - p1.cpp
 - p2.cpp
 - p3.cpp
- Deberá subir sus archivos directamente a www.gradescope.com.
- Recuerde que Gradescope solo conserva el último envío que se realiza. Si va a modificar su entrega, debe volver a adjuntar todos los archivos de su práctica.

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
 - Aplicar conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa. (Evaluar)
 - Analizar problemas e identificar los requerimientos computacionales apropiados para su solución. (Usar)
 - Usar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)
- Para los alumnos de las carreras de Ingeniería
 - Capacidad de aplicar conocimientos de matemáticas (nivel 3)
 - Capacidad de aplicar conocimientos de ingeniería (nivel 2)
 - Capacidad para diseñar un sistema, un componente o un proceso para satisfacer necesidades deseadas dentro de restricciones realistas (nivel 2)

Calificación:

Tabla de puntos (Sólo para uso del profesor)

Question	Points	Score
1	7	
2	6	
3	7	
Total:	20	

Preguntas

1. (7 puntos) Estructuras de control:

A continuación, se muestran los datos de vehículos asegurados por una compañía de seguros. Se muestra las horas al día que se desplazan dichos vehículos y la velocidad promedio con la que se desplazan:

Vehículo	Horas diarias (últimos 3 meses)	Velocidad promedio (últimos 3 meses)	Riesgo
001	3	50 km/h	Bajo
002	1	30 km/h	Bajo
003	8	99 km/h	Alto
004	9	80 km/h	Alto
005	2	15 km/h	Bajo
006	2	42 km/h	Bajo
007	8	90 km/h	Alto
008	1	41 km/h	Bajo
009	7	95 km/h	Alto
010	10	39 km/h	Bajo

También se ha etiquetado una característica asociada a dichos vehículos: Bajo riesgo y alto riesgo. Esta variable permite conocer qué vehículos tienen mayor riesgo de siniestralidad.

De manera general, los vehículos que se desplazan pocas horas al día y a una baja velocidad, tienen menor siniestralidad. En cambio, los vehículos que se desplazan muchas horas al día y a una alta velocidad, tienen mayor siniestralidad.

A los clientes de bajo riesgo, se les ofrecerá un descuento. Esta mañana, llegó un cliente preguntando si a él se le ofrecerá dicho descuento también. Sus datos son los siguientes:

Vehículo	Horas diarias (últimos 3 meses)	Velocidad promedio (últimos 3 meses)
011	0	120 km/h

Los ejecutivos de la aseguradora no saben cómo clasificar a este cliente. Maneja muy rápido, por lo que podría ser un cliente de riesgo. A 120 kilómetros de promedio, este conductor seguramente suele manejar a mayores velocidades y esto puede ocasionar accidentes.

Pero, dado que no suele mover su vehículo, podría ser rentable cobrarle el seguro todos los meses ya que, si no se moviliza, no hay ningún riesgo de que se choque. En los últimos meses, el promedio de horas al día que se ha desplazado es de 0 horas. Posiblemente algún día se desplazó por pocos minutos a un promedio de 120 kilómetros por hora y el resto de los días no movió el vehículo. Esto ocasiona que el promedio de horas al día sea cero.

Sin embargo, el día que por fin saque su vehículo a pasear, puede haber mucho riesgo asociado. Y puede sufrir un accidente, el cual la aseguradora tendría que cubrir.

Para clasificar a este cliente como «bajo riesgo» o «alto riesgo», se le comparará con el conductor más parecido a él. Se verificará si el conductores más parecidos a él es de bajo riesgo o de alto riesgo. Se clasificará al nuevo cliente en base a esa comparación.

Se le pide:

Elaborar un programa en lenguaje C que solicite como entrada las horas diarias y la velocidad promedio de un nuevo conductor, identifique cuál es el conductor más cercano a este y haga la clasificación tomando como referencia la categoría de riesgo de dicho vecino.

Por ejemplo, para el caso de este ejercicio, usted podrá ingresar 0 horas y 120 velocidad promedio. Pero la compañía de seguros desea que usted elabore un programa que permita clasificar a cualquier otro potencial cliente.

La clasificación se hará siempre en base a los 10 clientes iniciales. Es decir, después de haber clasificado al cliente número 11, se desea clasificar a un cliente número 12, dicha clasificación se hará también en base a los 10 clientes iniciales.

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100\% corresponde al puntaje indicado en cada punto.

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (100%).	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (70%).	El diseño tiene algunas deficiencias pero la ejecución es correcta (30%).	El diseño es deficiente y la ejecución no es correcta (0%).
Sintaxis	No existen errores sintácticos o de compilación (100%).	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (50%).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (30%).	El código tiene errores de sintaxis que afectan el resultado (10%).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (100%).	El código es de buen performance durante la ejecución (70%).	El código no está optimizado pero la ejecución no es deficiente (30%).	El código no está optimizado y la ejecución es deficiente (0%).

2. (6 puntos) Funciones:

Felicitaciones, al gerente de cumplimiento normativo le ha gustado el programa que usted hizo en la pregunta 1. Sin embargo, está preocupado porque hacer una comparación con el vecino más cercano puede no ser la mejor forma de evaluar a los clientes nuevos.

Él desea que la oficina de cumplimiento normativo haga algunas simulaciones usando una cantidad variable de vecinos más cercanos.

Tal vez se puede usar a los 3 vecinos más cercanos o tal vez se puede usar a los 2 vecinos más cercanos, y solo usar al tercero solo en caso de desempate. La oficina de cumplimiento normativo desea hacer diferentes pruebas con el programa que usted hizo en el ejercicio 1 antes de tomar una decisión.

Se le pide:

Elaborar una función en lenguaje C que solicite una cantidad variable de vecinos más cercanos (1, 2 o 3) y permita clasificar a un nuevo cliente como de «bajo riesgo» o de «alto riesgo» en base a los 10 clientes iniciales.

Para los vecinos pares, es posible que la cantidad de vecinos de «bajo riesgo» sea igual a la cantidad de vecinos de «alto riesgo». En esa situación, su programa simplemente deberá clasificar al nuevo cliente como «alto riesgo».

Caso	Clasificación
Mayor número de vecinos de bajo riesgo que de alto riesgo.	Bajo riesgo
Igual número de vecinos de bajo riesgo que de alto riesgo.	Alto riesgo
Menor número de vecinos de bajo riesgo que de alto riesgo.	Alto riesgo

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100\% corresponde al puntaje indicado en cada punto.

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (100%).	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (70%).	El diseño tiene algunas deficiencias pero la ejecución es correcta (30%).	El diseño es deficiente y la ejecución no es correcta (0%).
Sintaxis	No existen errores sintácticos o de compilación (100%).	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (50%).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (30%).	El código tiene errores de sintaxis que afectan el resultado (10%).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (100%).	El código es de buen performance durante la ejecución (70%).	El código no está optimizado pero la ejecución no es deficiente (30%).	El código no está optimizado y la ejecución es deficiente (0%).

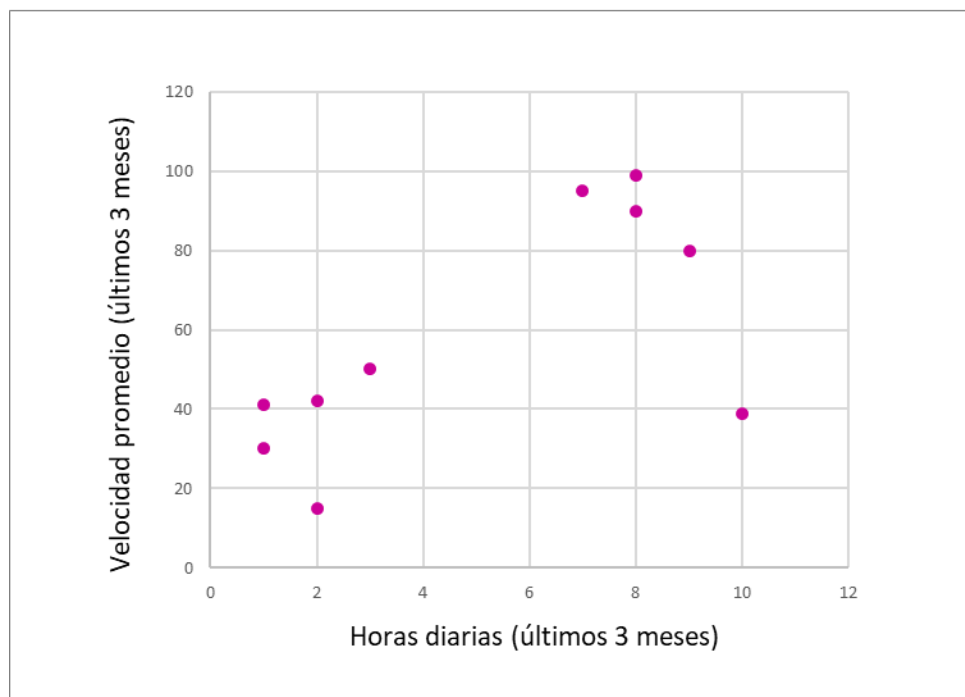
3. (7 puntos) punteros:

A continuación, se muestran los datos de vehículos asegurados por una compañía de seguros. Se muestran las horas al día que se desplazan dichos vehículos y la velocidad promedio con la que se desplazan:

Vehículo	Horas diarias (últimos 3 meses)	Velocidad promedio (últimos 3 meses)
001	3	50 km/h
002	1	30 km/h
003	8	99 km/h
004	9	80 km/h
005	2	15 km/h
006	2	42 km/h
007	8	90 km/h
008	1	41 km/h
009	7	95 km/h
010	10	39 km/h

Si graficáramos todos estos clientes en un gráfico de 2 dimensiones, podemos ver que abajo a la izquierda está el grupo de los clientes de bajo riesgo (manejan pocas horas al día y se desplazan a una velocidad moderada) y arriba a la derecha está el grupo de los clientes de alto riesgo (manejan muchas horas al día, y se desplazan a una mayor velocidad).

Cientes de seguro Vehicular



De manera general, los clientes de bajo riesgo están más cerca al origen $(0, 0)$ y los clientes de alto riesgo está más cerca al (∞, ∞) .

Sin embargo, el Gerente de Riesgos y Función Actuarial desea saber quién es el cliente de menor riesgo, y quién es el cliente de mayor riesgo. El cliente de menor riesgo será el más cercano al origen $(0, 0)$ y el cliente de mayor riesgo será el más lejano al (∞, ∞) . Para simular el punto (∞, ∞) usaremos las coordenadas $(1000, 1000)$.

Se le pide:

Elaborar una función en lenguaje C que calcule las coordenadas del cliente de menor riesgo y las coordenadas del cliente de mayor riesgo.

Esta función deberá recibir como parámetros al arreglo de horas, al arreglo de velocidad, y deberá calcular las coordenadas (x_1, y_1) del cliente de menor riesgo y las coordenadas (x_2, y_2) del cliente de mayor riesgo de tal forma que estas dos coordenadas puedan ser comunicadas hacia afuera de la función.

Desde la main, usted deberá llamar a dicha función, y enviar **desde adentro de la función**, la información correspondiente a las coordenadas (x_1, y_1) del cliente de menor riesgo y las coordenadas (x_2, y_2) del cliente de mayor riesgo **hacia main**.

Una vez que reciba las coordenadas de estos puntos en su función main, deberá imprimirlos.

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100\% corresponde al puntaje indicado en cada punto.

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (100%).	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (70%).	El diseño tiene algunas deficiencias pero la ejecución es correcta (30%).	El diseño es deficiente y la ejecución no es correcta (0%).
Sintaxis	No existen errores sintácticos o de compilación (100%).	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (50%).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (30%).	El código tiene errores de sintaxis que afectan el resultado (10%).

Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (100%).	El código es de buen performance durante la ejecución (70%).	El código no está optimizado pero la ejecución no es deficiente (30%).	El código no está optimizado y la ejecución es deficiente (0%).
--------------	---	--	--	---