

## Indicaciones generales

### Entrega:

- Cuenta con 3 horas (sincrónicas) para solucionar el examen.
- Debe desarrollar el examen (desde el inicio hasta fin) en el siguiente repositorio de Github classroom (<https://classroom.github.com/a/JvTuo18c>). No se aceptarán exámenes que no se hayan desarrollado en su totalidad en Github, en el repositorio respectivo.
- Una vez transcurridas las 3 horas tendrá 5 minutos adicionales para subir un archivo (.pdf) a Mediación Virtual que documente el hash (identificación del commit) y enlace al repositorio que contenga la versión final del examen. El enlace de mediación virtual se cerrará en punto. Los 5 minutos adicionales se otorgan para solventar cualquier problema técnico para la entrega del archivo. En caso de no entregar el archivo en Mediación Virtual, se considerará el examen sin entregar.
- La entrega final debe compilar, tal como está establecido en la carta al estudiante. De lo contrario, será calificado con nota cero.
- Debe subir el archivo del examen en Mediación Virtual al enlace correspondiente. En caso de problemas con la plataforma de Mediación Virtual debe mandar el examen por correo (sivana.hamer@ucr.ac.cr). En caso excepcional que fallen ambos medios, debe enviar la docente por medio de un mensaje privado en Telegram. Se debe subir el examen siempre durante el período de entrega del examen.
- Si realiza el examen en una computadora de escritorio que depende de energía eléctrica y sucede un fallo en el suministro, debe continuar su examen a papel. En caso de realizar el examen a papel, debe tomar fotografías de su solución. Toda página debe estar debidamente enumerada e incluir su identificación (e.g., cédula) en las fotografías. Debe subir el examen siempre durante el período de entrega del examen.

### Desarrollo:

- Para el examen es obligatorio que se conecten por medio de Zoom con un dispositivo que tenga cámara y audio (como indica la Resolución de la Vicerrectoría de Docencia No. VD-11502-2020). Debe compartir pantalla durante la realización de la prueba (salvo casos previamente definidos con la docente). Debe mantenerse en Zoom durante toda la duración del examen. Se tomará asistencia. Desconectarse de Zoom sin finalizar el examen y sin comunicarse con la docente puede provocar la anulación de la prueba.
- El examen es estrictamente individual. Está prohibido interactuar con cualquier otra persona que no sea la docente. En caso de sospechar lo contrario, se aplicará la política de plagio de la carta al estudiante.
- Durante el examen, la docente indicará cuando se deben realizar commits y el mensaje respectivo. Los commits no tienen que compilar. Realizar el commit correspondiente es obligatorio. Además de hacer commit al cambio tienen que realizar el push respectivo. El examen se considerará inválido si no viene con los commits-push solicitados y tendrá un cero como nota.
- Debe hacer uso de buenas prácticas para la programación orientada a objetos. Esto incluye pero no se limita a nombres significativos, indentación adecuada, clases con responsabilidades separadas, breve documentación de métodos y clases, y convenciones de nombres.

- Está prohibido utilizar librerías externas o instrucciones de programación no vistas en clases si la docente no lo ha indicado explícitamente. Su uso tendrá una penalización en la calificación final. Además, para este examen específicamente no puede utilizar la clase ArrayList.
- En este examen se permite la reutilización de código propio o aquel provisto en el curso (pero debe acreditarse la autoría). También se tiene derecho a consultar material escrito en internet. Todo código no realizado por él o la estudiante debe encontrarse debidamente referenciado.
- La comunicación entre docente y estudiante durante la realización de la prueba es por medio de Zoom. Para solicitar a la docente, de clic en el botón de "Pedir ayuda".
- Cualquier imprevisto técnico durante el examen deberá comunicarlo inmediatamente a la docente por medio de un mensaje privado en Telegram.
- Si normalmente utiliza el internet de la casa con señal wifi y se va la electricidad, pero posee internet en el celular comuníquese inmediatamente a la docente por un mensaje privado de Telegram.

### Recomendaciones:

- Antes de comenzar a implementar código, tenga total claridad sobre cada pregunta y solo responda lo que se le pida, ya que responder otros aspectos no permite otorgarle puntuación alguna. Ejemplo: si no se les pide leer, usen constantes, si no se les pide usar el patrón MVC, no tienen por qué hacerlo.
- No se olviden de guardar el código antes de subirlo (en Visual Studio Code es: ctrl + s).
- Si posee una computadora portátil, es una buena alternativa que la utilice para realizar el examen, asegurándose de que la batería está cargada al 100 %, para que en caso de una falla eléctrica pueda continuar el examen. Tengan disponible siempre el cargador.
- Garanticen que su teléfono está con la batería cargada o tiene disponible el cargador.
- Eviten distracciones. Vayan al baño y a comer antes del examen. Intenten encontrarse en una zona con silencio. Avisen a sus familiares que van a realizar un examen.

## Enunciado

La robótica es un campo interdisciplinario que busca el diseño, construcción y uso de máquinas denominadas robots para que hagan labores que hacen los seres humanos. Una de las compañías más destacadas en el campo de robótica es Boston Dynamics, quienes se enfocan en crear robots que son altamente dinámicos y móviles. Algunos de los robots que tienen incluyen a Spot, Stretch y Pick. Los robots que han creado en la compañía han sido usados para operaciones de rescate, transporte de materiales y inspecciones remotas de ambientes peligrosos.

Actualmente la compañía se encuentra creando un nuevo sistema para movilizar el robot. Este nuevo sistema de movilización va a permitir que, a partir de un mapa, un robot se pueda mover hasta llegar a la meta. La meta será una bandera que indica la última posición donde se tiene que mover el robot.

**Generación del mapa (10 %).** Al iniciar, el programa va a solicitar un tamaño  $N$  del mapa.  $N$  debe ser un número positivo (mayor a 0). Dado el  $N$ , se va a crear un mapa de tamaño  $N \times N$  de dos dimensiones. Cada posición del mapa puede contener a un *elemento*.

Un ejemplo de cómo se solicitará el tamaño es el siguiente.

Digite el n del mapa: 5

**Generación de los elementos del mapa (25 %).** Al comenzar el programa también se le debe indicar dónde se va a encontrar los *elementos* del mapa. La posición inicial en el mapa de cada elemento es dada por la fila y columna, que se le darán al programa. Las filas y las columnas son representadas por un número que va de 1 a  $N$ .

Un elemento tiene un símbolo que es el carácter que se va a mostrar en el mapa, una descripción que detalla lo que representa ese elemento y una cantidad de posiciones que se ha movido. En nuestro caso, tenemos dos elementos, un robot con un símbolo O y una bandera que representa la meta con el símbolo X. Se debe validar que al crear el elemento se de una posición válida en el mapa. Todos los elementos inician en la cantidad de posiciones que se movieron en cero.

Un ejemplo de cómo se solicitará las posiciones de los elementos del mapa es el siguiente.

Digite la fila del robot: 4

Digite la columna del robot: 1

Digite la fila de la bandera: 1

Digite la columna de la bandera: 5

En el caso del ejemplo, se va a crear un robot en la fila 4 y columna 1. Mientras, que se crea una bandera en la fila 1 y columna 5.

**Imprimir el mapa (10 %).** Se va a poder imprimir el mapa siguiendo el formato que sale en el ejemplo. En un mapa 5x5 se ve de la siguiente manera.

	1	2	3	4	5
1					X
2					
3					
4		O			
5					

Viene una cabecera para las filas y las columnas. El símbolo O (fila 4 y columna 2) representa al elemento de robot. El símbolo X (fila 1 y columna 5) representa a la bandera de la meta.

**Mostrar el menú de opciones (10 %).** El programa después de crear el mapa y los elementos, con su validación va a mostrar un menú de opciones. El menú deberá repetirse hasta que termine el programa. Además, deberá imprimir el mapa cada vez antes de las opciones. Las opciones del menú son las siguientes.

- **W.** Permite mover el robot hacia adelante un espacio.
- **A.** Permite mover el robot a la izquierda un espacio.
- **S.** Permite mover el robot hacia atrás un espacio.
- **D.** Permite mover el robot a la derecha un espacio.
- **X.** Termina el programa.

**Movimiento de los elementos (35 %).** Al digitarse cualquiera de las opciones de movimiento (W, A, S o D) debe moverse en la dirección respectiva el robot. Es decir, mover el robot debe actualizar donde se encuentra el mapa (no se duplica). Cada vez que se logra hacer un movimiento válido se le suma uno a la cantidad de posiciones que se ha movido el elemento. El movimiento funciona de la siguiente manera:

- **Mover arriba (W).** Se mueve el robot a la fila de arriba de la posición actual. Usando la posición del ejemplo donde se generan los elementos del mapa, si el robot se mueve hacia arriba se va a encontrar en la fila 3 y columna 2.
- **Mover izquierda (A).** Se mueve el robot a la columna de la izquierda de la posición actual. Usando la posición del ejemplo donde se generan los elementos del mapa, si el robot se mueve a la izquierda se va a encontrar en la fila 4 y columna 1.
- **Mover atrás (S).** Se mueve el robot a la fila de abajo de la posición actual. Usando la posición del ejemplo donde se generan los elementos del mapa, si el robot se mueve hacia arriba se va a encontrar en la fila 5 y columna 2.
- **Mover derecha (D).** Se mueve el robot a la columna de la derecha de la posición actual. Usando la posición del ejemplo donde se generan los elementos del mapa, si el robot se mueve a la derecha se va a encontrar en la fila 4 y columna 3.

Si al mover el elemento del robot, no puede moverse en la dirección, se va a quedar en el mismo espacio. Por ejemplo, si el robot quiere moverse hacia arriba y se encuentra en la fila 1 y columna 2, se va a quedar en la posición donde estaba. Pero, si puede moverse hacia la izquierda, atrás o derecha.

**Fin del programa (10 %)** . El programa puede terminar en los siguientes casos: el elemento del robot llega al elemento de la bandera y si se digita la opción X en el menú de opciones. Debe mostrar en el programa un mensaje cuando se termine el programa que incluya la cantidad de posiciones que se movió el robot antes de finalizar el juego.

### Notas:

- Su solución debe utilizar el paradigma orientado a objetos, siguiendo buenas prácticas.
- Debe utilizar arreglos ([]) en el examen. No puede usar ArrayList.
- Los datos que se le dan al programa en los ejemplos de entrada y salida están resaltados en **negrita**.