

**Instrucciones:**

- ◊ Fecha de publicación: 9 de febrero de 2021 a las 8:00 am.
- ◊ Fecha de entrega: 9 de febrero de 2021 hasta las 8:30 am.
- ◊ Medio de entrega: <https://e-aulas.urosario.edu.co> (no se reciben entregas por correo electrónico u otros medios).
- ◊ La actividad **debe** realizarse **individualmente**.
- ◊ **Formato de entrega:** Archivo de texto `.txt` conteniendo el pseudocódigo completo que es solución al problema del enunciado.

**Protocolo para la evaluación:**

Los siguientes lineamientos serán seguidos de forma estricta y sin excepción.

1. Solamente será posible tener acceso a [e-aulas.urosario.edu.co](https://e-aulas.urosario.edu.co) y a los sitios web correspondientes a la documentación de Python y C++ dispuestos por el profesor.
2. Maletas, morrales, bolsos, etc. deben estar ubicados al frente del salón.
3. Celulares y otros dispositivos electrónicos deben estar apagados y ser guardados dentro de las maletas antes de ser ubicadas en su respectiva posición.
4. El estudiante no debe intentar ocultar ningún código que no sea propio en la solución a la actividad.
5. El estudiante solo podrá disponer de hojas en blanco como borrador de apuntes (opcional).
6. El estudiante puede tener una hoja manuscrita de resumen (opcional). Esta hoja debe estar marcada con nombre completo.
7. E-aulas se cerrará a la hora en punto acordada para el final de la evaluación. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de e-aulas será calificado tal como está. Si ningún tipo de material es entregado por este medio, la nota de la evaluación será 0.0.  
**Se aconseja subir a e-aulas versiones parciales de la solución a la actividad.**
8. Todas las evaluaciones serán realizadas en el sistema operativo GNU/Linux.
9. Todas las entregas están sujetas a herramientas automatizadas de detección de plagio en códigos.
10. La evaluación debe presentarse exclusivamente en uno de los computadores ubicados en el salón de clase y a la hora acordada. Presentar la evaluación desde otro dispositivo o en otro horario diferente al estipulado es causa de anulación.

No habrán excepciones a estas reglas.

**Problem statement:**

Solve the following problem based on what was discussed during the lectures, and your own work, on line-segment properties and the convex hull problem. You may write pseudocode using the notation followed in class, or an enumerated brief description of the steps followed by the algorithm.

1. A simple polygon is a polygon that does not intersect itself and has no holes. The convex hull of a simple polygon is the convex polygon of minimum perimeter that contains a given simple polygon. See the Figure below.



Figure: The convex hull of a simple polygon (blue) is the whole region shaded in either color (blue or yellow).

Consider a simple polygon with  $n$  vertices, devise and describe an algorithm that computes the convex hull of a simple polygon in  $O(n)$  time. You may use any data structures you deem appropriate. Consider that a simple polygon is represented as a list  $P$  of counterclockwise-ordered points  $(x, y)$  in the plane.

Use the following pseudocode template to construct your algorithm.

```
1  SIMPLE_POLY_CONVEX_HULL(P):  
2  Input: A set P of points in the plane in counterclockwise  
        order (anything else?)  
3  Output: A list L containing the vertices of CH(P) in  
         counterclockwise order (anything else?)  
4  BEGIN  
5      ...  
6      ...  
7      ...  
8  
9      RETURN L  
10 END
```