

*#1-Crea un programa que*

*#Reciba Nombres de alumnos hasta que escriba "Fin"*

*#Guarde esos nombres en una lista.*

*#Muestre la lista completa y cuantos alumnos se registraron.*

*#2-Simule una pila de navegación en un navegador web*

*#Inserta 3 paginas web "Google.com", "wikipedia.org", "Openai.com".*

*#El usuario presiona "Volver", una vez.*

*#Muestra la pagina actual (la del tope de la pila)*

*#3-Simula una cola de impresion:*

*#Los documentos llegan en este orden: "Doc1.pdf", "Tarea.docx", "Presentacion.pptx."*

*#Atiende (elimina) el primer documento en la cola.*

*#Llega un nuevo documento "informe.pdf"*

*#Muestra el orden de la impreson final.*

*#4- Tienes una lista de 10,000 estudiantes ordenadas por nombre ¿que algoritmo de búsqueda usarías para encontrar un estudiante específico y por que?*

*Elegiría el de Merge en  $O(n^2)$  ya que ahí se puede realizar una mejor búsqueda y solo encontrar el estudiante en específico*

*#5- Explica con un ejemplo practico por que la diferencia entre  $O(n)$  y  $O(n^2)$  es importante en aplicaciones reales.*

*Por que unos programas pueden ser lentos ósea cuando hay muchos datos estos pueden ser muy tardados en  $O(n^2)$  y que el  $O(n)$  se ejecuta de manera mas eficiente y se requieren en situaciones específicas.*

*#6- Tienes que ordenar una lista de 100 numeros que casi esta ordenada ¿que algoritmo elegirías y porque?*

*Yo elegiría el  $O(1)$  ya que es una lista ya casi ordenada y usando bubble.*

*#7-¿ hasta que valor de n podrias calcular Fibonacci usando la version recursiva antes de que sea demasiado lento? ¿y con la version iterativa?*

*Mucho antes de 50,000 por que ahí el programa ya es demasiado pesado, y con la iterativa como 10,000 ya que es una tarea un poco mas sencilla*

*#8 Diseña un sistema que use diferentes estructuras de datos y algoritmos para optimizar estas operaciones*

*#Búsqueda de usuarios por ID (1,000,000 usuarios).*

*#Ordenamiento de productos por precio (10,000 productos).*

*#Calculo de Estadísticas en tiempo real.*