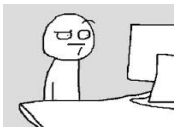


Universidad del País Vasco Euskal Herriko Unibertsitatea

Prog. Básica - Laboratorio 9
Listas dinámicas

Nombre:_____**Fecha**:_____

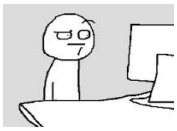
AVISOS:

1. Cuando uno de los ejercicios que se proponen carece de programa de prueba o de plantilla, significa que lo tienes que hacer desde cero. Además, el hecho de que se proporcionen algunos casos de prueba no significa que no pueda faltar alguno. ¡Tienes que añadir los casos de prueba que falten!

2. Tienes que subir a eGela los ficheros fuente (.adb) comprimidos en un fichero (.zip) que deberá ajustarse a las reglas de nombrado exigidas hasta ahora (por ejemplo, EJauregi_lab7.zip).

3. Se presupone que los ejercicios son correctos, es decir, que no tienen errores de compilación y que funcionan correctamente. Esto significa que las soluciones de los ejercicios no puntúan nada por ser correctas, pero penalizan si tienen errores o no se ajustan a lo que se pide. Una vez que la solución es correcta, lo que se evalúa (lo que puntúa) son:

- **Los casos de prueba: ¿Se han contemplado todos los casos de prueba, desde los generales a los más críticos? Se dan algunos, pero faltan muchos otros.**
- **La eficiencia: ¿Se utilizan los "chivatos" cuando hace falta? ¿No hay asignaciones innecesarias? ¿No hay condicionales que no deberían ejecutarse?, ¿Se definen solamente los parámetros o variables necesarios?, etcétera.**
- **La claridad: ¿El código está tabulado? ¿Los nombres de las variables ayudan a entender el código? ¿Hay un único *return* al final de la función?, etcétera.**



1. Número de elementos de una lista

Dada una lista de enteros, especificar e implementar un subprograma que calcule el número de elementos de la lista.

Plantillas ofrecidas:

longitud.adb	Para hacer el subprograma en Ada
prueba_longitud.adb	Para completar y probar el subprograma en Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb, esc.adb (No hay que modificarlo).

2. Media aritmética de los valores de una lista

Dada una lista de enteros, especificar e implementar un subprograma que calcule la media aritmética de los elementos de la lista.

Plantillas ofrecidas:

media.adb	Para hacer el subprograma en Ada
prueba_media.adb	Para completar y probar el subprograma en Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb, esc.adb (No hay que modificarlo).

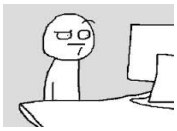
3. Máximo de la lista y posición

Dada una lista de enteros, especificar e implementar un subprograma que obtenga el elemento máximo de la lista y su posición.

Plantillas ofrecidas:

calcular_maximo_y_posicion.adb	Para hacer el subprograma en Ada
prueba_calcular_maximo_y_posicion.adb	Para completar y probar el subprograma en Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb, esc.adb (No hay que modificarlo).



4. Insertar al final

Dada una lista de enteros y un valor entero, especificar e implementar un subprograma que inserte ese valor detrás del último elemento de la lista.

Plantillas ofrecidas:

insertar_al_final.adb	Para hacer el subprograma en Ada
prueba_insertar_al_final.adb	Para completar y probar el subprograma Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, esc.adb (No hay que modificarlo).

5. Buscar en lista no ordenada

Dada una lista de enteros **no ordenada** y un valor entero, especificar e implementar un subprograma que diga si el valor pertenece o no a la lista. En caso de que el valor pertenezca a la lista devolverá su posición y si no devolverá cero.

Plantillas ofrecidas:

posicion.adb	Para hacer el subprograma en Ada
prueba_posicion.adb	Para completar y probar el subprograma en Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb, esc.adb (No hay que modificarlo).

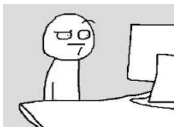
6. Buscar en lista ordenada

Dada una lista de enteros **ordenada** (de menor a mayor) y un valor entero, especificar e implementar un subprograma que diga si el valor pertenece o no a la lista. En caso de que el valor pertenezca a la lista devolverá su posición y si no devolverá la posición en que debería colocarse.

Plantillas ofrecidas:

posicion_lista_ordenada.adb	Para hacer el subprograma en Ada
prueba_posicion_lista_ordenada.adb	Para completar y probar el subprograma en Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb, esc.adb (No hay que modificarlo).



7. Insertar en lista ordenada

Dada una lista de enteros **ordenada** y un valor entero, especificar e implementar un subprograma que inserte el valor en el lugar apropiado.

Plantillas ofrecidas:

insertar_en_lista_ordenada.adb	Para hacer el subprograma en Ada
prueba_insertar_en_lista_ordenada	Para completar y probar el subprograma Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, esc.adb (No hay que modificarlo).

8. Insertar en la posición N-ésima

Dada una lista de enteros, un valor positivo N y un valor entero, especificar e implementar un subprograma que inserte el entero en la posición N de la lista. (no hay programa de prueba)

Plantillas ofrecidas:

insertar_en_posicion_N.adb	Para hacer el subprograma en Ada
?????????.adb	Para completar y probar el subprograma en Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb , esc.adb (No hay que modificarlo).

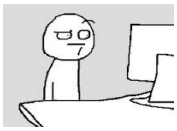
9. Borrar la primera aparición

Dada una lista de enteros y un valor entero, especificar e implementar un subprograma que borre ese elemento de la lista.

Plantillas ofrecidas:

eliminar_primera_aparicion.adb	Para hacer el subprograma en Ada
prueba_eliminar_primera_aparicion.adb	Para completar y probar el subprograma en Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb , esc.adb (No hay que modificarlo).



10. Borrar todas las apariciones

Dada una lista de enteros y un valor entero, especificar e implementar un subprograma que borre todas las apariciones de ese elemento de la lista.

Plantillas ofrecidas:

eliminar_todas_las_apariciones.adb	Para hacer el subprograma en Ada
prueba_eliminar_todas_las_apariciones.adb	Para completar y probar el subprograma Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb , esc.adb (No hay que modificarlo).

11. Intersección de dos listas

Dadas dos listas de enteros, especificar e implementar un subprograma que obtenga una lista con los elementos que aparecen en las dos listas.

Plantillas ofrecidas:

interseccion.adb	Para hacer el subprograma en Ada
prueba_interseccion.adb	Para completar y probar el subprograma Ada

Fichero adicional: datos.ads, crear_lista_vacia.adb, ins.adb , esc.adb (No hay que modificarlo).

Nota: podéis usar el subprograma **posicion.adb** que habéis implementado anteriormente.

12. Actualizar lista

Implementa en Ada el procedimiento *actualizar_lista* que, a partir de una lista dinámica de números ordenados ascendentemente L, una posición I y una cantidad C, modifique el elemento i-ésimo de L aumentándolo en una cierta cantidad C, dejando de nuevo la lista igualmente ordenada. Si la posición no existe, se debe crear un nuevo elemento y suponer que su valor anterior era 0.

Plantillas ofrecidas:

actualizar.adb	Para hacer el subprograma en Ada
prueba_actualizar.adb	Para completar y probar el subprograma Ada