2)

a) Verdadero, las instrucciones dentro de una estructura “repeat until” siempre se van a ejecutar al menos una vez, debido a que en esta estructura primero se ejecutan las instrucciones y después de evalúa la condición.

b) Falso, que el programa este modularizado no quiere decir que sea eficiente, para que lo sea debe cumplir con lo requerido en el menor tiempo posible.

c) Verdadero, por más que posea todos sus campos del mismo tipo, el registro es una estructura de datos heterogénea.

d) Verdadero, puede haber un puntero que apunte a un vector y este estaría almacenado en la memoria dinámica.

e) Falso, los elementos de una lista se almacenan en la memoria dinámica, la estructura de datos lista es una estructura dinámica, es decir se puede agregar elementos según el algoritmo lo requiera siempre que la memoria lo permita y por lo tanto la misma varia en la ejecución del programa.

f) Falso, si es posible, aunque de una manera un poco rebuscada, sabiendo la cantidad de elementos, es decir recorrer la lista con un mientras e ir contando la cantidad de elementos en una variable. Después utilizar esta variable con un for que vaya de 1 hasta la cantidad dada e ir imprimiendo cada elemento.

g) Verdadero, una variable global puede ser pasada como parámetro real en la invocación de un módulo debido a que las variables globales pueden ser utilizadas en todo el programa, tanto dentro como fuera de los módulos.

h) Falso, un pseudocodigo no se realiza para un lenguaje de programación en especifico, se realiza para explicar la lógica y los pasos en general de un algoritmo.

3) Lo primero que hay que hacer es recorrer el vector, guardar la primer posición de las repeticiónes del elemento a eliminar en una variable (pos), obtener la cantidad de repeticiones del valor a eliminar recibido por parámetro (repetir 1 hasta dimL, si (v[i]=elem) entonces sumar 1 a cant (cant:=cant+1)). Hacer un corrimiento de la cantidad de elementos restantes luego de las repeticiones, hacia pos (for i:= 0 to dimL-fin-1) do v[pos+i] := v[fin+i]. Por ultimo decrementar la dimL(dimL-cant).

4)

MEMORIA ESTATICA:

dimF: 4 bytes

v: 4\*50 = 200bytes

e: 11+11+8 = 30bytes

i: 4bytes

suma: 8bytes

max: 8bytes

4+200+30+4+8+8 = 254bytes.

MEMORIA DINAMICA:

30\*50 = 1500bytes (peor de los casos)

5)

TIEMPO DE EJECUCION

T1 = 1 + 1 = 2ut

T2 = 50+1\*3+5+50\*5 = 408ut

Repeat until: (n + 1) \* 3(condiciones) + cuerpo + n\*cuerpo

T3 = 50 + 1 + 50\*5 = 301ut

While: n + 1(ev. condicion) + n\*cuerpo

T1 + T2 + T3 = 2 + 408 + 301 = 711 UT.