1)

Calcular: 175, 0 , 175

C = 20

A = 100

B = 0

Writeln: 15, 20 ,175

A = 15

B= 20

C = 20

2)

a) Falso, las instrucciones dentro de una estructura de control for no siempre se ejecuta al menos una vez, ya que se ejecuta según el índice que le indiquemos, y si este es 0, no se va a ejecutar.

b) Falso, un programa modularizado no quiere decir que sea eficiente, para que lo sea tiene que cumplir con lo requerido en el menor tiempo de ejecución y memoria ocupada posible.

c) Falso, el acceso a los campos de un registro no necesita respetar el orden en el que están declarados, debido a que son una estructura de datos a la cual se puede acceder directamente, simplemente indicando el campo al cual queramos acceder.

d) Falso, una variable global puede ser accedida y modificada en todo el programa, tanto dentro como fuera de los módulos.

e) Falso, la variable de tipo puntero se va a poder utilizar, nada mas que no vamos a reservar espacio en la memoria si no hacemos el new(p).

f) Verdadero, en todos los campos de un registro puede entrar un valor y devolver otro.

g) Falso, las listas son un tipo de estructura de datos homogénea, es decir, todos sus elementos son de un mismo tipo.

h) Falso, ya que “NIL” no libera la memoria referenciada, sino que corta la conexión entre un puntero y la memoria referenciada, pero ese espacio queda ocupado e inaccesible. Para poder liberar la memoria referenciada se debe usar “DISPOSE”.

3)Un vector es un conjunto de datos indexados (ordenados por un indice), es tipo de estructura de datos homogénea, es decir todos sus elementos son del mismo tipo. Se puede acceder directamente a la posición que queramos simplemente indicando la posición. Es una estructura estática, debido a que la memoria que ocupa se define en la creación del vector y la cantidad de memoria es fija, no varía durante la ejecución del programa. También en el mismo se maneja una dimensión física, y una dimensión lógica.

Para insertar en una posición determinada lo primero que hay que hacer es validar la posición, verificar si hay espacio en el arreglo, hacer corrimiento de los valores hacia la derecha (a partir de la dimL con un for i:= dimL downto pos do v[i+1]:= v[i]), asignar el valor, aumentar la dimL. (!!!).

4) MEMORIA ESTATICA:

const dimF = 2 bytes.

v: vector = 10 \* 4(puntero) = 40

e: emple = 20 bytes

i: integer = 2 bytes

2 + 40+ 20 + 2 = 64bytes

MEMORIA DINAMICA:

Repeat until: 20 \* 10 (peor de los casos) = 200bytes

5) TIEMPO DE EJECUCION:

T1 = 1UT

T2 = 10 +1 + 3 + 3\* 10 = 44UT

Repeat until = n +1(ev. Condición) + cuerpo + cuerpo \* n

T3 = 10 + 1 + 5 \* 10 = 61 UT

While = n + 1(ev. Condición) + cuerpo \* n

T1 + T2 + T3 = 106UT