Conceptos y paradigmas de lenguajes de programación – Practica 3

1. a) Tome una de las variables de la línea 3 de código e indique y defina cuáles son sus atributos  
   Variable ‘a’:  
   Nombre: a  
   Alcance: 3-16  
   Tiempo de vida: 1-16  
   L-valor: Automática  
   R-valor: 0,1,2,3,4,5,6,7,8,9

b) Compare los atributos de la variable del punto anterior con los atributos de la variable de la línea 4. ¿Que dato contiene esta variable?  
id: p  
l-valor: Automática  
r-valor: Indefinido  
Alcance: 4-16  
Tiempo de vida: 1-16  
  
id: p^  
l-valor: Dinámico  
r-valor: Indefinido  
Alcance: 4-16  
Tiempo de vida: 1-16

1. a)  
   Inicialización por defecto  
   Inicialización por declaración  
     
   b)
2. Explique los siguientes conceptos asociados al atributo l-valor:

• Variable estática: son sensibles a la historia. El l-valor de una variable estatica se asigna en tiempo de compilación, tiene una direccion fija y conocida desde el principio en una zona de memoria especial reservada para las variables estaticas y globales. El tiempo de vida comienza antes del bloque de declaracion, antes de la alocacion, antes de la ejecución. Su valor no varia durante la vida del proceso en ejecución. Normalmente tiene un ambito mas amplio que otras variables. Los valores se pueden establecer una vez (durante el tiempo de ejecución) o se pueden cambiar en múltiples ocasiones durante la ejecución del programa. Son usadas en C. Conserva su valor cuando se vuelve a llamar a la fucion que la contiene. Se inicializan solo una vez (por defecto igual a 0).

• Variable automática o semiestática: (*automatica: cuando se aloca en memoria la unidad que lo contiene ¿?*) se crea en tiempo de compilación y se destruye al terminar la ejecución del modulo donde fueron declaradas. Aparecen en la declaración. Su almacenamiento en memoria se asigna automáticamente según su alcance. Cuando termina su alcance el compilador libera esa posición de memoria perdiéndose su valor. Se usa en pascal. Ejemplo:

procedure uno (n: integer);

var x: integer; (cada vez que se invoca uno, se vuelve a declarar e inicializar)

• Variable dinámica: (*es creada por una instrucción explicita y se desaloca por una instrucción explicita*)creada mediante declaraciones escritas en el codigo, cuando se elabora la declaracion al iniciarse la ejecución de la region donde se encuentra y se destruye cuando termina dicha ejecución. Se crean usando la palabra reservada new y se devuelve la direccion del bloque de memoria asociada a la variable que se crea, que se alamacena en una variable puntero, para poder acceder posteriormente a la variable dinamica. Se usa en ADA.

Ejemplo:

p:= new Integer; (p es un puntero, un acceso a variables de tipo integer).

• Variable semidinámica: no se sabe en tiempo de compilación cuanto lugar ocupara en la memoria, sino en ejecución. Una vez que se establece su tamaño no varía. Se usa en Delphi, Ada.

Ejemplo:

Var arrayD: array of string;

Set length (arrayD, 10); (reserva espacio para el elemento)

)  
  
\* Investigue si C utiliza variables automáticas y dinámicas. ¿Utiliza otro tipo de almacenamiento? ¿Cuales? Explique  
  
C usa variables automaticas, dinamicas y estaticas. El modificador *auto* se usa para definir el ambito temporal de una variable local. Es el modificador por defecto para las variables locales (*auto int r*). El modificador *static* se usa con el fin de que las variables locales de una funcion conserven su valor entre distintas llamadas a la misma (*static int n*).

1. A) ¿A qué se denomina variable local y a qué se denomina variable global?  
   Una variable local es aquella cuyo ambito se restringe a la funcion que la ha declarado, es local a esa funcion. Esta variable solo puede ser referenciada en esa función. Una variable global se define fuera de cualquier función al principio del programa principal. Su ámbito son todas las funciones que componen al programa. Se puede referenciar en cualquier parte del programa.  
     
   B) ¿Una variable local puede ser estática respecto de su l-valor? En caso afirmativo de un ejemplo  
   Las variables locales (declaradas dentro de una funcion) son automaticas por defecto (solo existen mientras se ejecuta la funcion). Cuando se invoca la funcion se crean estas variables en la pila y se destruyen al finalizar la funcion. La unica excepcion son las variables locales declaradas como estaticas (*static*), donde la variable mantiene su valor entre cada dos llamadas a la funcion aunque su visibilidad sigue siendo local a la funcion. Entonces, una variable estatica declarada dentro de una funcion trabaja como una variable automatica, con la diferencia que como ocupa un lugar permanente y fijo en la memoria, guarda su valor de llamada en llamada.

C)  
D)