|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lenguaje** | **Tipo de Variable** | **L-Valor** | **R-Valor** | **Notas a considerar** |
| C | Global numérica | Automática | No es indefinido, toma valor por defecto | En C, si se compilan varios archivos, cada uno tendrá sus globales y aplicarán las normas. Esto también afecta al tiempo de vida de las variables. |
|  | Global puntero | Automática | No es indefinido – Null por defecto | C no inicializa las variables locales, solo globales y estáticas. |
|  | Global estática | Estático | No es indefinido – Toma valor por defecto (si tiene) |  |
|  | Local estática numérica | Estático | No es indefinido – Por defecto |  |
|  | Local declarada tipo: main(){int i=4} | Automático | Indefinida |  |
|  | Puntero local | Automático | Indefinida |  |
| ADA | Constante numérica del tipo c: constant integer := 10 | Automático | Indefinido, es constante común | ADA parece siempre inicializar en Null los punteros. |
|  | Constante numérica del tipo c: constant := 10 | Automático | Definido, liga valor en compilación (en este caso 10) |  |
|  | Arreglo del tipo (0..10) | Automático | Indefinido |  |
|  | Arreglo del tipo (0..c) donde c **está** definida del estilo:  “c:constant integer:=10” | Semi-dinámica | Indefinido |  |
|  | Arreglo del tipo (0..c) donde c **está** definida (ej. c: constant := 10) | Automático | Indefinido |  |
|  | Puntero | Automático | Null |  |

Los arreglos en C son de alocacion Automatica? si

Semantica estatica es sintaxis?

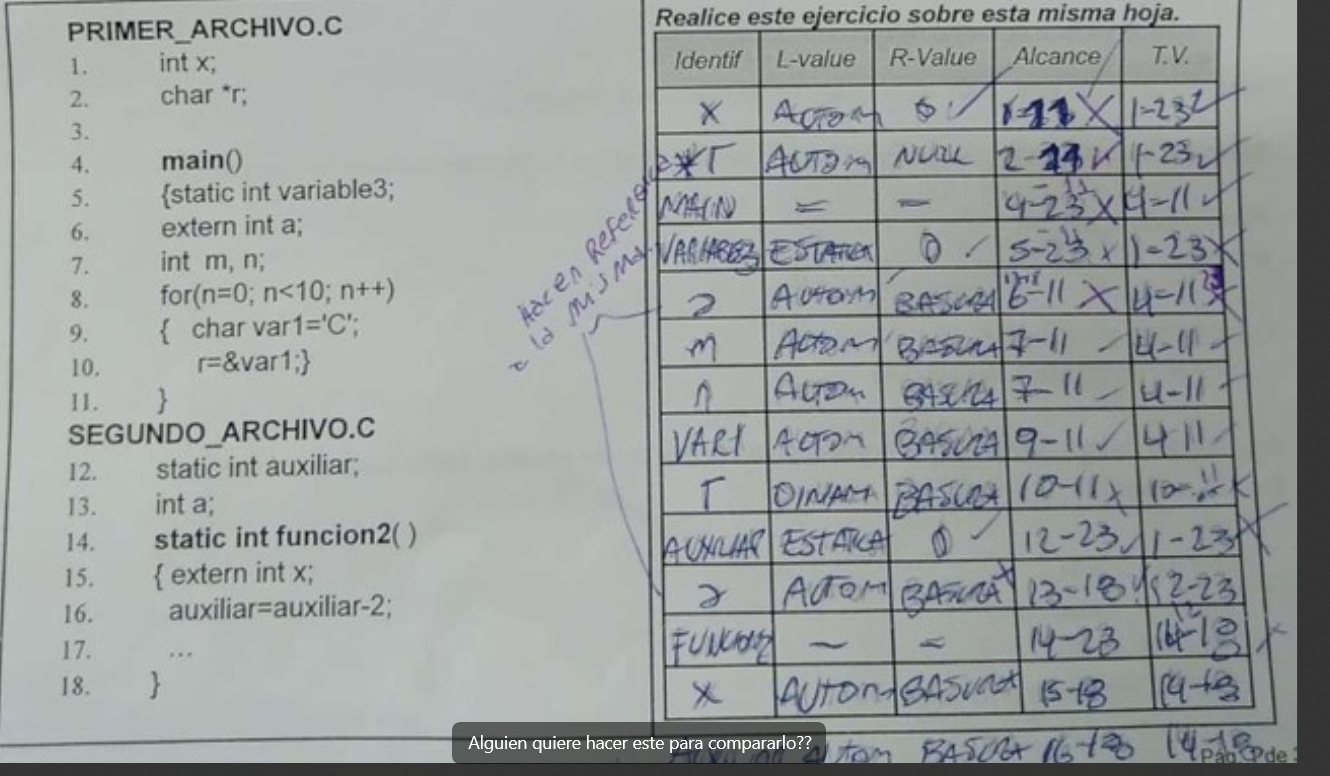
Como es el R-valor de \*r? Es “C”? Y si var1 no se hubiera inicializado en la asignación seria indefinido o basura? Es basura

El R-valor de var1 es “C”?

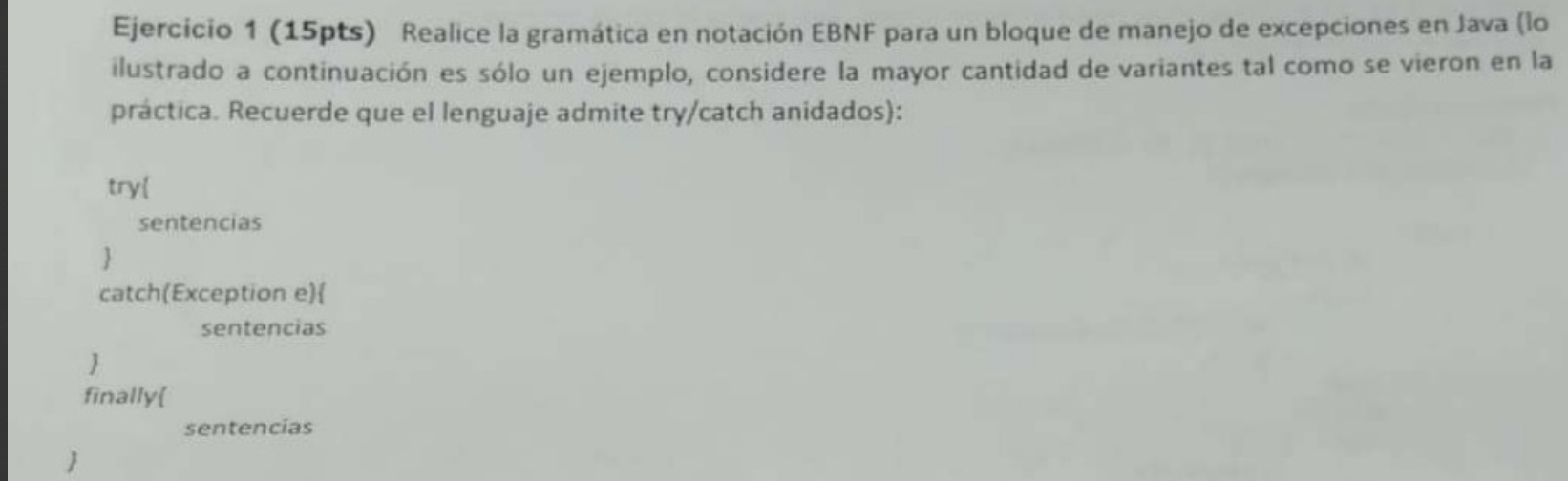
Cual es el tiempo de vida de auxiliar? No debería ser todo el programa porque es estatica?

El tiempo de vida de funcion2 es 14-23? Por que? Influye en que sea estatica?

Si un procedimiento llama a 2 funciones que le devuelven un valor cada uno, la 2da función pisa el VR de la primera función?



Como resolver esto.



En diagrama de Conway:

