##### Ejercicio integrador 2016 ­ Parte I

Sobre el los lenguajes asignados, (los dos lenguajes definidos para el grupo al que pertenece):

1. Realice una comparación respecto de los criterios de evaluación de los lenguajes de programación asignados. Mencione al menos dos criterios.
2. Identifique las características principales de la sintaxis y realice la gramática en EBNF de una variable en ambos lenguajes asignados.
3. Enuncie las las diferencias existentes respecto de la asignación e inicialización por defecto de variables entre ambos lenguajes. Ejemplifique.
4. Determine los tipos de variable de acuerdo al momento de ligadura con el l­valor con que cuenta cada lenguaje. Ejemplifique la forma en que simularía (en caso de no poseerla) una variable de tipo estática. En el caso en que cuente con este tipo de variables ejemplifique la diferencia con una variable global. Utilice el compilador de los lenguaje asignados que se encuentra en la siguiente página <http://www.tutorialspoint.com/codingground.htm>​ para realizar los ejemplos planteados.

**IMPORTANTE:**

**La entrega se debe realizar únicamente en un archivo pdf,** el cual debe contener las siguientes

partes:

1­-  Una carátula o encabezado con el nombre y legajo de cada uno de los integrantes del grupo y el número de grupo asignado por la cátedra y los lenguajes asignados, así como la bibliografía consultada. No se puede referenciar wikipedia. Las referencias deben ser de páginas oficiales del lenguaje, de libros, de manuales del lenguaje, de documentación de cátedras de enseñanza del lenguaje, de papers/artículos, entre otros.

2­- El enunciado y desarrollo de la tarea pedida (código incluido). El código debe ser correcto para poder probarlo en ​[http://www.tutorialspoint.com/codingground.htm​.](http://www.tutorialspoint.com/codingground.htm)

**El archivo entregado debe tener el siguiente nombre grupoXX.pdf** (respetar este formato de nombre)  donde xx representa el nro. de grupo asignado por la cátedra. Sólo debe ser subido a la plataforma por uno de los integrantes del grupo y debe tener como máximo, un total de 5 carillas. Se recomienda realizar un ejercicio por carilla.

1. Analizando algunos criterios de evaluación de los lenguajes de programación otorgados por la cátedra:

Los lenguajes de programación deberían poder producir programas fáciles de escribir y de leer, y resultar fáciles a la hora de aprenderlo o enseñarlo, es decir, que sean **simples y legibles,** además deser **eficientes**, entre otras características.

A la hora de analizar el lenguaje C, notamos que la **legibilidad** se ve deteriorada por el lenguaje al proporcionar varias alternativas para la especificación del mismo concepto; es un lenguaje de nivel medio por lo que atenta contra la **simplicidad** del mismo**.** Dando más de una forma para denotar un mismo concepto, se incrementa el tamaño del lenguaje y se favorece el desarrollo de “dialectos” que usan solamente subconjuntos de tales formas. Un programador experto en un dialecto puede tener dificultades para la comprensión de programas escritos en un dialecto diferente.

**LEGIBILIDAD ESTRUCTURA DE CONTROL MATLAB: INDENTACION**

Uno de los ejemplos que ofrece C para visualizar el problema mencionado anteriormente, es la suma de 1 a la variable entera ‘**a**’, donde notamos que se puede efectuar con cualquiera de las siguientes sentencias:

1. a++; **2)** a=a+1; **3)** a +=1; **4)** ++a;

Entre otras cosas, las restricciones estrictas en la longitud de los identificadores, pueden forzar al programador a dar nombres crípticos a las variables y procedimientos.

Por el contrario, existe un entorno de computación y desarrollo de aplicaciones para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos, aplicando el lenguaje de alto nivel Matlab, que tiene como característica la **simplicidad,** ya que se utiliza para realizar programas breves y **simples**; osea que trabajar con Matlab comporta aprender un lenguaje simple.

Una gran cantidad de métodos numéricos se encuentran implementados de una forma muy **eficiente** y son accesibles como simples comandos. De esta forma, Matlab se puede utilizar como una caja negra: el usuario pregunta y el ordenador responde sin que éste tenga que preocuparse de qué tipo de operaciones se han efectuado por el camino.

La especificación y asignación de cada variable la realiza Matlab en forma dinámica y **eficiente**, por lo que no son necesarias las declaraciones de variables de algunos lenguajes de computación convencionales.

C también es apreciado por la **eficiencia** del código que produce; es un lenguaje potente y eficiente, permitiendo obtener programas rápidos y compactos.

**2**      Un programa escrito en C debe tener el siguiente formato general:

main ()

{Sentencia1;

   Sentencia2;

          .

}

*Las constantes se declaran de la siguiente manera: #define <nombre> <valor>; donde el #define es una directiva para el compilador que indica que se reemplazará por en el texto. Ejemplo: #define max\_valor 50.*

*Con respecto a las estructuras de control:*

*                       Condicionales: if\_else: if (condición) inst1 [else inst2 ]*

*                       Ciclos:*

*     While:  while (exp) inst;*

*     For:       for (contador ; condicion; variacion ) inst;*

*     Do\_while: do inst; while (cond);*

***¿Es necesario?***

 Funciones:

 En C no existen procedimientos, sólo hay funciones. Los procedimientos pueden ser simulados por medio de funciones que no retornan ningún valor. <tipo\_ret>  <nombre> (<parámetros>) { declaraciones (es opcional); instrucciones; };donde<tipo\_ret> es el tipo de valor que devuelve la función (por defecto int); para indicar que no se devuelve nada se coloca void.<parámetros> es una lista de elementos, separados por comas. Para devolver el valor se tiene la instrucción return.

La sintaxis de declaración de una variable es la siguiente:

tipo nombre;   
  Donde tipo es cualquier tipo válido de C; y nombre es un identificador válido en C, esto es, una letra seguida opcionalmente por otras letras y números; no se admiten signos diacríticos (acentos, tildes, etc.) y que las letras mayúsculas y minúsculas no se consideran equivalentes. Por otra parte, no se admite utilizar como identificadores los nombres de palabras reservadas en C.

La sentencia for de MATLAB es muy diferente y no tiene la generalidad de la sentencia for de C:

         for contador = valorInicial : valorFinal;

sentencias del cuerpo del bucle

end

Los valores iniciales y finales, pueden ser constantes o variables.

         Sentencia if-else: if condición

                                 sentencias del bloque;

                            else

                                  sentencias del bloque;

                            end

         Sentencia while:

                      while condición

                           sentencias del bloque;

                       end

Matlab no requiere ningún tipo de declaración de variables sino que, una vez que se utiliza una variable, Matlab crea la respectiva variable reservando el espacio de memoria necesario.

    Con respecto a la asignación de una variable: nombre = tipo, donde el tipo puede ser: entero, real, complejo, carácter, etc., que es asignado automáticamente por el lenguaje; Ejemplos: a = 5;a = a\*a;

**3)**

**En C**

La asignación por defecto de variables en el lenguaje de programación C es distinta para cada tipo de variable que puede utilizarse.

Si la variable es una variable estatica, entonces se inicializa por defecto con el valor numérico cero. En caso de ser una variable no estática, el valor por defecto de la variable es *indeterminado*.

El valor de una variable indeterminado no se puede predecir. En la mayoría de los casos tienen valor sin sentido, pero existen raras ocasiones en que la variable posee un valor de otra variable antigua que estaba almacenada en la misma región de memoria que C reservó.

Ejemplo:

void func() {

**static** int x; 🡪 inicializa por defecto en 0

int f; 🡪 valor indeterminado

printf("%d**\n**", x);

x = x + 1;

}

**En Matlab**

La asignación por defecto de una variable es un concepto que no existe en Matlab. Esto se debe a que las variables no es necesario declararlas sino que existen en el momento en que se le asignan un valor.

La ventaja de esto es que no se va a tener que lidiar con variables no asignadas, aunque esto no significa que no se vaya a tener problemas con variables incorrectamente inicializadas.

Ejemplo:

*puntos = 235;*

*pi = 3.14;*

Bibliografiaa

Artículo:

https://archuto.files.wordpress.com/2011/02/clase902.ppt

libro Ghezzi

http://www.monografias.com/trabajos94/matlab-y-sus-comandos/matlab-y-sus-comandos.shtml

http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/5/13-507-2-DR.pdf

material ofrecido por la teoría de la materia cydp.

http://decsai.ugr.es/~jfv/ed1/c/cdrom/cap1/f\_cap12.htm

<http://iimyo.forja.rediris.es/tutorial/intro.html>

<http://stackoverflow.com/questions/1597405/what-happens-to-a-declared-uninitialized-variable-in-c-does-it-have-a-value>

http://matlabxsiemprexd.blogspot.com.ar/2007/11/para-crear-una-variable-en-matlab.html