DEL PSEUDOCÓDIGO AL LENGUAJE DE PROGRAMACIÓN C

En la siguiente tabla, se muestran algunos ejemplos de equivalencias entre lo que se hace en pseudocódigo y cómo se debe hacer en C.

DESCRIPCIÓN	PSEUDOCÓDIGO	C	OBSERVACIONES
Operadores Aritméticos			
Suma	+	+	
Resta	-	-	
Multiplicación	*	*	
División	/	/	
Módulo	%	%	
Operadores Lógicos			
Conjunción	Λ	&&	
Disyunción	V	||	✓Esta barra aparece al lado izquierdo del 1
Negación	~	!	
Operadores Relacionales			
Mayor que	>	>	
Menor que	<	<	
Mayor o igual	>=	>=	
Menor o igual	<=	<=	
Igual	=	==	✓Solo para comparaciones (no asignación)
Diferente	<>	!=	
Definición de Variables			
Numéricas Enteras	Entero numero1, numero2	**int** numero1, numero2;	✓Siempre debe terminar en *;*
Numéricas Reales	Real resultado	**float** resultado; **double** resultado;	✓*float* para reales pequeños ✓*double* para reales grandes ✓Siempre deben terminar en *;*
Caracter	Caracter letra Caracter nombre[30]	**char** letra; **char** nombre[30];	✓Siempre debe terminar en *;*

DESCRIPCIÓN	PSEUDOCÓDIGO	C	OBSERVACIONES
Instrucciones y Estructuras			
Entrada de datos	Lea: numero Lea: division Lea: letra Lea: nombre	scanf ("%d",&numero); scanf ("%f",&division); scanf ("%c",&letra); scanf ("%s",&nombre);	✓ *%d* para enteros ✓ *%f* para reales pequeños ✓ *%c* para un caracter, *%s* para cadena de caracteres ✓ Siempre debe terminar en *;*
Salida de datos	Escriba: "El resultado de la suma de ", num1, "y", num2, "es =", sum	printf ("El resultado de la suma de %d y %d es = %d", num1, num2, sum);	✓ Las variables siempre van al final separadas por coma. ✓ Donde debe aparecer el valor de la variable en el mensaje, se coloca el *%d*, *%f* o *%s*, dependiendo del tipo de dato. ✓ Siempre debe terminar en *;*
Asignación	sum←num1+num2	sum = num1+num2;	✓ Se reemplaza ← por el =, teniendo en cuenta que es una asignación y no una comparación. ✓ Siempre debe terminar en *;*

Condicionales simples	Si (a>b) ent
	  //otras instrucciones
	Fsi
		if (a>b)
		{
		  //otras instrucciones;
		}
			✓ Ni el *if* ni las *{ }* llevan *;*

Condicionales dobles	Si (a>b) ent
	  //otras instrucciones
	sino
	  //otras instrucciones
	Fsi
		if (a>b)
		{
		  //otras instrucciones;
		}
		else
		{
		  //otras instrucciones;
		}
			✓ Se cumple las mismas reglas para los condicionales simples.
			✓ Debe haber *{ }* para el *if* y otro para el *else*

DESCRIPCIÓN	PSEUDOCÓDIGO	C	OBSERVACIONES

Selección Caso	SeleccionCaso (variable_control)
	Caso 1: //otras instrucciones
	salga
	
	// ……
	
	Caso 9: //otras instrucciones
	salga
	
	OtroCaso: //instrucciones por defecto
	Fcaso
		switch (variable_control)
		{
		case 1: *//otras instrucciones;*
		break;
		
		// ……..
		
		case 9: *//otras instrucciones;*
		break;
		
		default: *//instrucciones por defecto;*
		}
			✓ El *break* debe terminar en *;*
			✓ Ni el *switch* ni las *{ }* llevan *;*

Ciclo Mientras	con ← 1
	Mq (con < = n) haga
	//otras instrucciones
	con= con + 1
	FMq
		con = 1;
		while (con <= n)
		{
		//otras instrucciones;
		con= con + 1; *//con++;*
		}
			✓ Ni el *while* ni las *{ }* llevan *;*

Ciclo Haga Mientras	con ← 1
	Haga
	//otras instrucciones
	con= con + 1
	Mq (con <= n)
		con = 1;
		do
		{
		//otras instrucciones;
		con = con + 1; *//con++;*
		
		} **while** (con < n);
			✓ Cuando hay un *do while* , este *while* es la única vez que lleva *;*
			✓ Ni el do ni las *{ }* terminan en *;*

Ciclo Para	Para (con ← 1, n , 1) haga
	//otras instrucciones
	Fpara
		for (con=1 ; con <= n ; con++)
		{
		//otras instrucciones;
		}
			✓ Ni el *for* ni las *{ }* llevan *;*