|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Karina García Morales |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1121 |
| *No de Práctica(s):* | Práctica no. 9 |
| *Integrante(s):* | Espinosa Carrillo Luis Andrés |
|  |  |
| *No. de Equipo de cómputo empleado* | Equipo no. 20 |
| *Semestre:* | 2019-1 |
| *Fecha de entrega:* |  |
| *Obervaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Guía de práctica de estudio 09:**

**Estructuras de repetición**

**Objetivo:**

Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva def.

**Desarrollo:**

Las estructuras de repetición son las llamadas estructuras cíclicas, iterativas o de bucles. Permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que la expresión lógica a evaluar se cumpla (sea verdadera).

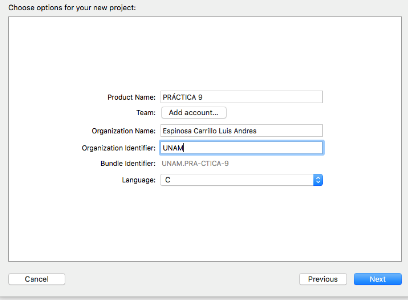
En lenguaje C existen tres estructuras de repetición: while, do-while y for. Las estructuras while y do-while son estructuras repetitivas de propósito general.

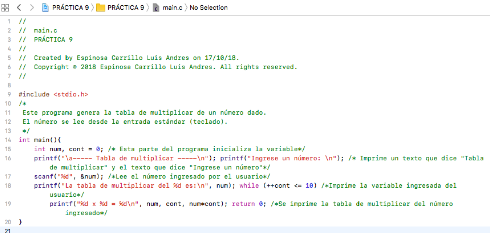
**Estructura de control repetitiva while**

La estructura repetitiva (o iterativa) while primero valida la expresión lógica y si ésta se cumple (es verdadera) procede a ejecutar el bloque de instrucciones de la estructura, el cual está delimitado por las llaves {}. Si la condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura, es decir, el bloque se puede ejecutar de cero a ene veces.

Ciclo while no lleva ; ya que sólo contiene una instrucción.

Durante la realización de la práctica copiamos el programa de la práctica de la página 152 que realizaba la tabla de multiplicar de un número ingresado por el usuario, al principio sólo comentamos lo que hacía cada línea del programa:

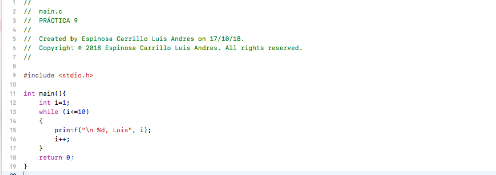




Y luego ejecutamos el programa:

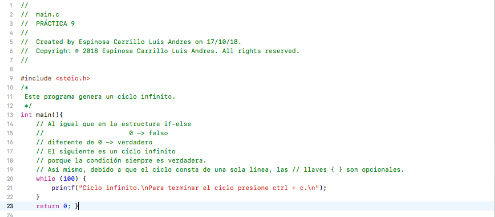


Hicimos un programa que nos imprimiera nuestro nombre y las veces que se imprimió usando un ciclo while:



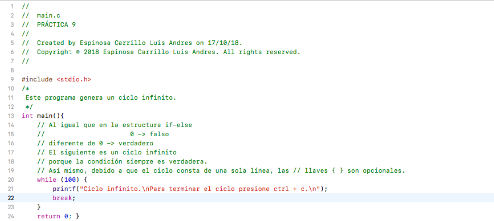


Copiamos el programa de la página 152 de la práctica para generar un ciclo infinito que nos imprime infinitamente el resultado:





Después le pusimos un break; para que sólo se imprimiera una sola vez la instrucción, este funciona cuando hay una condición de verdadero o falso.



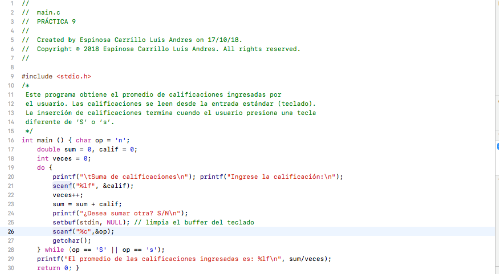


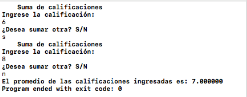
**Estructura de control repetitiva do-while**

do-while es una estructura cíclica que ejecuta el bloque de código que se encuentra dentro de las llaves y después valida la condición, es decir, el bloque de código se ejecuta de una a ene veces.

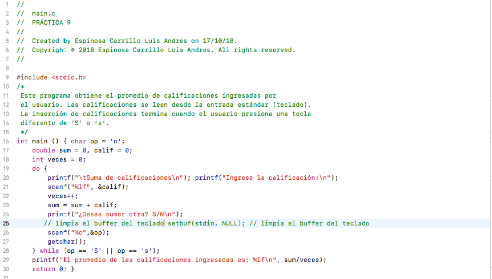
La estructura de control do-while **SÍ** lleva ;

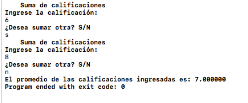
Copiamos el programa de la página 153 en dónde al ingresar el número nos pedía ingresar más números para generar un promedio:





Luego comentamos el comando setbuf:



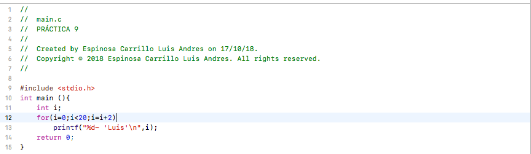


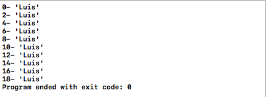
**Estructura de control de repetición for**

Lenguaje C posee la estructura de repetición f***or l***

La estructura *for*ejecuta 3 acciones básicas antes o después de ejecutar el bloque de código. La primera acción es la inicialización, en la cual se pueden definir variables e inicializar sus valores; esta parte solo se ejecuta una vez cuando se ingresa al ciclo y es opcional. La segunda acción consta de una expresión lógica, la cual se evalúa y, si ésta es verdadera, ejecuta el bloque de código, si no se cumple se continúa la ejecución del programa; esta parte es opcional. La tercera parte consta de un conjunto de operaciones que se realizan cada vez que termina de ejecutarse el bloque de código y antes de volver a validar la expresión lógica; esta parte también es opcional.

Hicimos un programa con el ciclo for para que nos imprimiera nuestro nombre sólo con los primeros números pares:



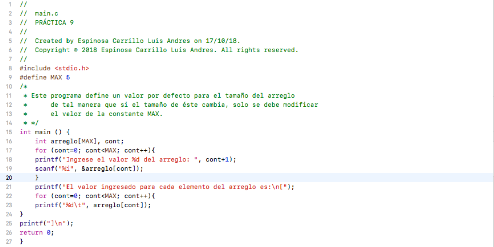


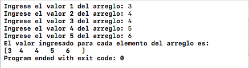
**Define**

Las líneas de código que empiezan con # son directivas del preprocesador, el cual se encarga de realizar modificaciones en el texto del código fuente, como reemplazar un símbolo definido con #define por un parámetro o texto, o incluir un archivo en otro archivo con #include.

*define*permite definir constantes o literales; se les nombra también como constantes simbólicas.

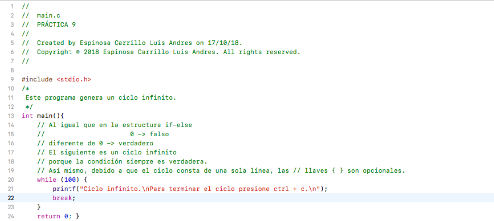
Realizamos un programa que nos imprimiera el valor para cada elemento del arreglo al nosotros ingresar el valor 6 del arreglo:





**Break**

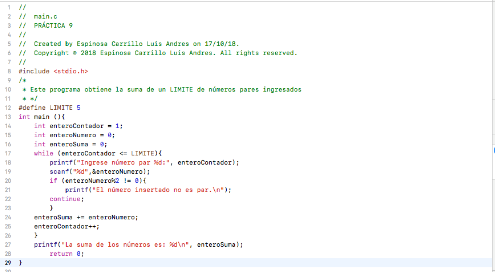
Algunas veces es conveniente tener la posibilidad de abandonar un ciclo. La proposición b**reak p**roporciona una salida anticipada dentro de una estructura de repetición, tal como lo hace en un switch. Un br*eak pr*ovoca que el ciclo que lo encierra termine inmediatamente.

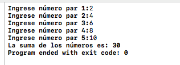




**Continue**

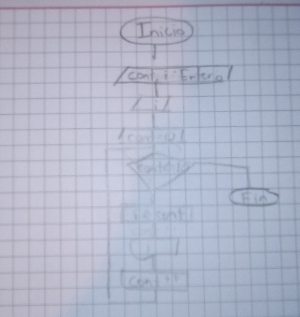
La proposición continue provoca que inicie la siguiente iteración del ciclo de repetición que la contiene.



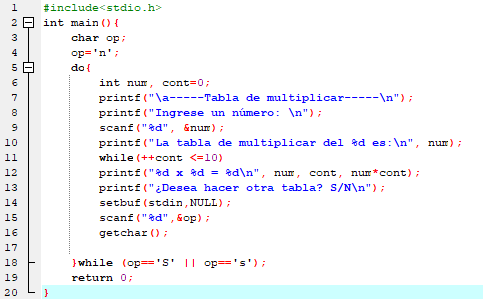


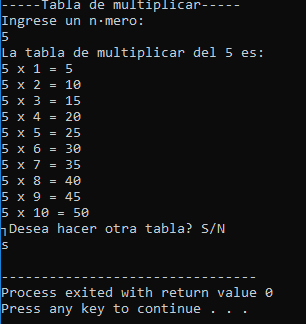
**Ejercicios de tarea**

1.- Dibujar el diagrama de flujo del programa para obtener la tabla de multiplicar con ciclo While.



2.- Cambiar el ejercicio de la calculadora, página 5 y agregar la pregunta de la opción si desea calcular o no (do{}while (op == 'S' || op == 's');)



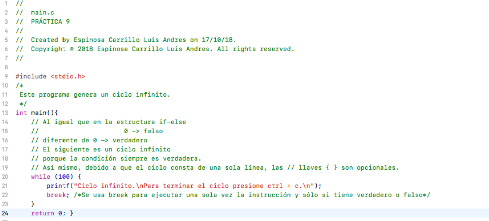


3.- Explica que sucede con el ciclo while al colocarle un valor positivo, negativo y cero y porque razón se obtiene ese resultado.

Esto ocurre porque en lenguaje máquina y en tabla de verdad, todo número diferente de 0 tiene como valor verdadero.

0 Es el único valor que equivale como falso.

4.- ¿Qué sucede con el ejercicio que ejecuta el break, si se omite y que sucede si se coloca?



Si se omite, el programa coree una infinidad de veces

5.- ¿Qué sucede con el ejercicio que ejecuta continue, si se omite y que sucede si se coloca?

Si se coloca, esta permita la siguiente iteración del ciclo, si se omite, no se ejecuta la siguiente iteración del ciclo

**Conclusiones**

No tuve complicación al realizar la práctica, los temas vistos durante la realización de la práctica en clase ya están bien entendidos y también los nuevos temas como el uso del break o del continuo en un ciclo.

**Referencias**

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.