

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingenieria en Computacion

Programación Estructurada

Actividad 6

ALUMNO: Isai Alexis Arredondo Urbalejo

MATRÍCULA: 368747

GRUPO: 932

PROFESOR: Nuñez Yepi Pedro

Introducción

La programación es un campo estrechamente ligado a la automatización y repetición de tareas. En este contexto, los ciclos se destacan como elementos fundamentales que permiten ejecutar un conjunto de instrucciones de manera continua según ciertas condiciones o un número predefinido de veces. En el lenguaje de programación C++, tres tipos principales de ciclos son ampliamente utilizados: el ciclo "while", el ciclo "for" y el ciclo "do-while". Estos ciclos ofrecen una versatilidad significativa para abordar diversos problemas y optimizar la eficiencia del código.

El ciclo "while" establece un escenario en el que un bloque de código se ejecuta mientras una condición dada sea verdadera, lo que proporciona una flexibilidad excepcional para situaciones en las que no se conoce de antemano el número de iteraciones necesarias. Por otro lado, el ciclo "for" se destaca cuando se tiene un conocimiento preciso de la cantidad de repeticiones requeridas, permitiendo una estructura de control concisa y legible. Finalmente, el ciclo "do-while" presenta una variante única al garantizar que el código se ejecute al menos una vez antes de evaluar la condición, siendo útil en escenarios donde se requiere una ejecución inicial incondicional.

Competencia

Comprobar el uso de los diferentes ciclos utilizados en el mismo contexto, en este caso los ejercicios propuestos por el profesor

Fundamentos

Ciclo for

En español se traduce como "ciclo para" y es uno de los bucles más utilizados en la programación gracias a su estructuración, ya que permite determinar el número de repeticiones a realizar (además, es necesario conocer dicho valor para su uso). El mismo está compuesto por cuatro parámetros:

- Variable de inicio del ciclo: es donde se comenzará a ejecutar.
- Condición o límite del ciclo: sirve como determinante de la continuidad o finalización del ciclo. Solo se repite en caso de que la condición sea verdadera.
- Modificador: se encargará de modificar la variable cuando sea necesario.
- Cuerpo: donde se añaden las instrucciones a repetir, pudiendo ser una o varias

Ciclo while

El ciclo de programación while (que se traduce como “ciclo mientras”) permite establecer una condición donde se repetirán las instrucciones hasta que dicha condición no se cumpla.

A diferencia del ciclo anterior, este no cuenta con un valor inicial o final, solo está compuesto por la condición. Sin embargo, es necesario añadir una sentencia que finalice el ciclo.

Este bucle se utiliza cuando se desea repetir ciertas instrucciones un número indeterminado de veces hasta que no se cumpla la condición.

Ciclo do-while

Se traduce como “repetir mientras”. A diferencia del bucle anterior, el bucle do-while primero ejecuta las instrucciones y luego hace la comprobación de la condición especificada, lo que significa que se ejecutará al menos una vez. En otras palabras, las acciones se realizarán hasta que se cumpla la condición.

Un claro ejemplo es un programa en el que se debe ingresar una cantidad determinada (digamos que del 1 al 5). Si el usuario añade un número fuera de dicho rango la aplicación le seguirá pidiendo un número.

Admin, & Admin. (2019). Ciclos de programacion. *Explicacion .Net*.

<https://www.explicacion.net/ciclos-de-programacion/>

Procedimiento

ACTIVIDAD 6

PARTE 1

REALIZA LOS SIGUIENTES EJERCICIOS EN C SUBIR UN PROGRAMA QUE LLAME LOS 3 EJERCICIOS Y CON CADA UNA DE LAS SENTENCIAS.

MENÚ

1.- FIBONACCI

2.- FACTORIAL

3.- CANTIDAD DE DÍGITOS

1.- PROGRAMA QUE PREGUNTE LA CANTIDAD DE VECES QUE DESEA QUE SE REALICE EL PROGRAMA DE FIBONACCI

2.- PROGRAMA QUE PIDA UN NÚMERO Y DESPLEGAR LA SALIDA DE FACTORIAL DE UN NÚMERO DADO.

EJEMPLO

$5 * 4 = 20$

$20 * 3 = 60$

$60 * 2 = 120$

FACTORIAL DE 5 = 120

3.- PROGRAMA QUE PIDA UN NÚMERO Y DESPLEGAR LA CANTIDAD DE DÍGITOS QUE TIENE EL NÚMERO.

EJEMPLO

25 TIENE 2 DÍGITOS

2578 TIENE 5 DÍGITOS

(NOTA REPETIR LOS PROGRAMAS CON `for()` , `while()` , `do{ }while()`)

PARTE 2

REALIZA LOS SIGUIENTES EJERCICIOS EN C SUBIR UN PROGRAMA QUE LLAME LOS EJERCICIOS

1.- El profesor de una materia desea conocer la cantidad de sus alumnos que no tienen derecho al examen de nivelación.

Diseñe un programa en C que lea las calificaciones obtenidas en las 5 unidades por cada uno de los 40 alumnos y escriba la cantidad de ellos que no tienen derecho al examen de nivelación (Los promedios menores de 50 no tiene derecho a examen de nivelación) .

2.- Realiza una función en C que imprima las tablas de multiplicar del 1 al 10

EJEMPLO:

TABLA DEL 1

$1 * 1 = 1$

$1 * 2 = 2$

$1 * 10 = 10$

PRESIONA UNA TECLA PARA CONTINUAR.

NOTA: Utilizar limpiar pantalla, esperar tecla

3. Realizar función en C para un programa que sirva para leer n cantidad de números dentro de un rango dado por el usuario, desplegar la suma de los números y la media aritmética de los números válidos dentro del rango.

4- En los cabos la embarcación finisterra que tiene fondo de cristal, solo sale a navegar con un máximo de 10 turistas o un máximo de 700 kilos de pasajeros. (preguntar el peso a cada turista) con un máximo de 15% de sobrepeso. desplegar el promedio de peso de los turistas y cual de los 2 condiciones se cumplió.

5.- Un alumno solo puede cursar la misma asignatura en un máximo de 3 veces, si el alumno reprueba durante sus intentos deberá repetir la materia, y si en su tercera ocasión no aprueba se le dará de baja académica.

Elabore una función donde basada en sus 3 exámenes parciales calcular el promedio y basado en su promedio final, se deberá enviar mensaje al alumno de repetir materia, aprobado o baja temporal.

NOTA: LOS PROBLEMAS 100% VALIDADOS..

Resultados y Conclusiones

Con las funciones vacías pude mandar a llamar diferentes programas del código. Con esto logre tener un código más ordenado y entendible, esto aunado a los ciclos vistos anteriormente en los fundamentos, pude probar diferentes maneras de hacer un mismo código pero usando cada una de los ciclos. Así me quedo mas claro como usar cada uno y me di cuenta que siempre hay forma de hacer el mismo programa pero de diferente manera.