



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 3:

Estructuras de Control

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 30 de Agosto del 2023

Introducción:

En esta práctica principalmente se aborda el tema de estructuras de control, esto es importante ya que nos permite a nosotros como programadores nos ayuda a tomar decisiones y controlar la ejecución del programa.

Las 3 principales estructuras son

- La secuencial
- La condicional
- En bucles

Estas estructuras de control forman la base de la programación y son esenciales para resolver una amplia variedad de problemas. Combinando estas estructuras de manera efectiva.

Nosotros como programadores podemos crear programas sofisticados que realicen tareas complejas y respondan a situaciones cambiantes de manera inteligente.

Competencia:

Diseñar programas de cómputo, aplicando las estructuras de control de selección, para proporcionar soluciones óptimas a problemas del área de ingeniería, de manera innovadora y ordenada.

Fundamentos:

En este capítulo se revisan los distintos métodos con los que C controla el flujo lógico de un programa.

Los operadores relaciones binarios que se usan son:

< Menor que

<= Menor Igualque

> Mayor que

>= Mayor igual que

== Exactamente Igual

!= Diferente

Los operadores lógicos binarios:

|| OR

&& AND

y el operador lógico unario de negación !, que sólo toma un argumento.

! NOT

Los operadores anterior son usados con las siguientes estructuras que se muestran.

La sentencia if

Las tres formas como se puede emplear la sentencia if son:

if (condicion)

sentencia;

...0

if (condicion)

sentencia1;

else

sentencia2;

...0

```
if (condicion1)
```

```
    sentencia1;
```

```
else
```

```
    if (condicion2)
```

```
        sentencia2;
```

```
    ...
```

```
else
```

```
    sentencian;
```

El flujo lógico de esta estructura es de arriba hacia abajo. La primera sentencia se ejecutará y se saldrá de la estructura if si la primera condición es verdadera. Si la primera condición fue falsa, y existe otra condición, se evalúa, y si la condición es verdadera, entonces se ejecuta la sentencia asociada.

Si existen más condiciones dentro de la estructura if, se van evaluando éstas, siempre y cuando las condiciones que le preceden sean falsas.

La sentencia que está asociada a la palabra reservada else, se ejecuta si todas las condiciones de la estructura if fueron falsas.

Por ejemplo:

```
int main()
```

```
{
```

```
    int x, y, w;
```

```
    .....
```

```
    if (x>0)
```

```
    {
```

```
        z=w;
```

```
    .....
```

```
}  
else  
{  
z=y;  
.....  
}  
}
```

La sentencia switch

Aunque con la estructura if ... else if se pueden realizar comprobaciones múltiples, en ocasiones no es muy elegante, ya que el código puede ser difícil de seguir y puede confundir incluso al autor transcurrido un tiempo.

Por lo anterior, C tiene incorporada una sentencia de bifurcación múltiple llamada switch. Con esta sentencia, la computadora comprueba una variable sucesivamente frente a una lista de constantes enteras o de carácter. Después de encontrar una coincidencia, la computadora ejecuta la sentencia o bloque de sentencias que se asocian con la constante.

La forma general de la sentencia switch es:

```
switch (variable) {  
case constante1:  
  
    secuencia de sentencias  
    break;  
case constante2:  
  
    secuencia de sentencias  
    break;  
case constante3:
```

secuencia de sentencias

break;

...

default:

secuencia de sentencias

}

Donde la computadora ejecuta la sentencia default si no coincide ninguna constante con la variable, esta última es opcional. Cuando se encuentra una coincidencia, la computadora ejecuta las sentencias asociadas con el case hasta encontrar la sentencia break con lo que sale de la estructura switch.

Las limitaciones que tiene la sentencia switch ... case respecto a la estructura if son:

- Sólo se tiene posibilidad de revisar una sola variable.
- Con switch sólo se puede comprobar por igualdad, mientras que con if puede ser con cualquier operador relacional.
- No se puede probar más de una constante por case.

La forma como se puede simular el último punto, es no teniendo sentencias asociados a un case, es decir, teniendo

una sentencia nula donde sólo se pone el caso, con lo que se permite que el flujo del programa caiga al omitir las

Sentencias, como se muestra a continuación:

Switch (letra)

{

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

```
numvocales++;
```

```
break;
```

```
case ' ':
```

```
numesp++;
```

```
break;
```

```
default:
```

```
numotras++;
```


Procedimiento:

El tema de estructuras de control de selección y cómo se representan en algoritmos, diagramas de flujo, pseudocódigo y como código en un lenguaje de programación estructurado de, posteriormente se muestra ejemplos de problemas dónde para su solución es necesario la aplicación de estructuras de control de selección para la toma de decisiones.

1.- Algoritmo que lea 3 calificaciones calcule el promedio del alumno y desplegar:

Si $\text{prom} < 30$ Repetir

Si $\text{prom} \geq 30$ y $\text{prom} < 60$ extraordinario

Si $\text{prom} \geq 60$ y $\text{prom} < 70$ suficiente

Si $\text{prom} \geq 70$ y $\text{prom} < 80$ Regular

Si $\text{prom} \geq 80$ y $\text{prom} < 90$ bien

Si $\text{prom} \geq 90$ y $\text{prom} < 98$ muy bien

Si $\text{prom} \geq 98$ y $\text{prom} \leq 100$ excelente

Si $\text{prom} > 100$ Error en promedio

(OPTIMIZADO FORMA DE ARBOL)

2.- Algoritmo que lea 3 números y desplegar cuál número es del medio y su valor

3.- Algoritmo que lea 3 números y desplegar los 3 números en orden ascendente

Conclusiones:

Como ya lo hemos visto las estructuras de control son muy importantes ya que se toman decisiones del código.

Las sentencias condicionales como (if, else, switch) y los bucles (for, while, do-while), son esenciales para diseñar algoritmos eficientes y resolver una amplia variedad de problemas.

Estas estructuras son la base de la programación y son fundamentales para el desarrollo de aplicaciones y sistemas informáticos en todas las disciplinas de la informática. Por lo tanto, dominar las estructuras de control es un paso crucial en el camino hacia la programación eficiente y efectiva.

```
float calif1, calif2, calif3, promedio;
```

```
printf("Ingrese la primera calificación: ");
```

```
scanf("%f", &calif1);
```

```
printf("Ingrese la segunda calificación: ");
```

```
scanf("%f", &calif2);
```

```
printf("Ingrese la tercera calificación: ");
```

```
scanf("%f", &calif3);
```

```
promedio = (calif1 + calif2 + calif3) / 3.0;
```

```
printf("El promedio es el siguiente: %.2f\n", promedio);
```

```
if (promedio < 30)
```

```
{
```

```
    printf("Repetir\n");
```

```
} else
```

```
{
```

```
    if (promedio < 60)
```

```
{  
    printf("Extraordinario\n");  
} else
```

Se utiliza una serie de estructuras if y else anidadas para determinar en qué rango se encuentra el promedio y mostrar un mensaje correspondiente.

Anexo:

<https://drive.google.com/file/d/14VcCFku0u6SMGib2IyPSIBzQuxsrdh7X/view?usp=sharing>