



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 4:

Estructuras de Control

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 4 de Septiembre 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 iteración, 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.- Programa en C que use un menú para realizar las 4 operaciones básicas, donde el usuario introduce 2 números enteros y realizará la operación según sea su selección del menú.

MENÚ

1.- Suma

2.- Resta

3.- Multiplicación

4.-División

2.- Programa en C que use un menú para realizar conversiones de unidades de medida, donde el usuario selecciona una opción y realizará la opción según sea su selección del menú.

MENÚ

1.- cm a pulgadas

2.- cm a pies

3.- Km a millas

4.- Pulgadas a cm

5.- pies a cm

6.- millas a Km

3.- Programa en C que lea 6 números, desplegar el valor del número mayor.

4.- Programa en C que sirva para calcular el salario semanal de un trabajador donde se obtiene como dato de entrada las horas semanales trabajadas, el salario por hora.

El programa deberá calcular el salario normal, salario extra y salario total, considerando lo siguiente:

Jornada Normal de 40 horas.

El salario normal se considera las horas trabajadas menores o igual a la jornada normal

Salario extra se considera las horas trabajadas mayores a la jornada normal y se pagan dobles las primeras 9 y triples a partir de la décima hora extra

Nota: Desplegar todos los datos (Salario x hora, Horas Trabajadas, Salario normal, Salario extra y Salario Total)

5.- Programa en C que sirva para desplegar el Total de una llamada telefónica donde se pide como datos de entrada los minutos y el tipo de llamada, se cobra de la siguiente manera:

MENÚ

1.- Llamada Local \$3.00 sin límite de tiempo

2.- Llamada Nacional \$7.00 por los 3 primeros minutos y \$2.00 minuto adicional

3.- Llamada Internacional \$9.00 por los 2 primeros minutos y \$4.00 minuto adicional

Desplegar, Subtotal, Iva (16%) y Total.

6.- Programa en C que sirva para calcular el Total a pagar por consumo de agua, donde el dato de entrada son los M3 de agua consumidos, Tomar en cuenta que se cobra escalonada de la siguiente manera:

Rango1: 0 al 4 M3 \$50 x facturación sin importar cuánto consumió en este rango

Rango2: 5 a 15 M3 \$8.00 x M3

Rango3: 16 a 50 M3 \$10.00 x M3

Rango4: 51 M3 en adelante \$11.00 x M3

Nota: Desplegar SubTotal, Iva(16%), y Total a pagar.

7- En la materia programación estructurada se aplican 5 exámenes, calcular el promedio final de la materia donde la calificación menor de los exámenes se anula y el promedio se calcula en base a 4 exámenes.

Desplegar el promedio final.

8.- Programa en C que sirva para el juego del CHINCHAMPU (Piedra, Papel, Tijera) para 1 jugador y la computadora, (usar condición anidada)

9.- Programa en C que sirva para el juego del CHINCHAMPU (Piedra, Papel, Tijera) para 1 jugador y la computadora, (usar selección múltiple)

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/1iR_V88f8ij1l8wevw4sXol91TptKSwlN/view?usp=sharing