

	UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERIA ESCUELA DE COMPUTACIÓN
Ciclo I	Programación de Algoritmos Guía de Laboratorio No. 6 Estructura selectiva (switch)

I. RESULTADOS DE APRENDIZAJE

Que el estudiante:

- Aplique de forma correcta a la estructura selectiva (`switch`)
- Resuelva problemas, que requieran el uso de `switch`.
- Diseñe menús de varias opciones haciendo uso de la estructura `switch`.

II. INTRODUCCIÓN
La Toma de decisiones en Lenguaje C++

La estructura selectiva *switch* selecciona una de entre múltiples alternativas. Esta estructura es especialmente útil cuando la selección se basa en el valor de una variable simple o de una expresión simple denominada **expresión de control o selector**.

Sintaxis estructura switch en C++

```

switch(selector) {
    case etiqueta1:
        sentencias1;
        break;
    case etiqueta2:
        sentencias2;
        break;
    ...
    case etiquetan:
        sentenciasn;
        break;
    default: //Opcional
        sentencias;
}

```

El valor de esta expresión puede ser de tipo **int, bool o char**. No se permiten tipos numéricos decimales (`float`, `double`).

A continuación se muestra la sintaxis de la estructura `switch` en el lenguaje de programación C++.

La expresión de control o **selector** se evalúa y se compara con cada una de las etiquetas de **case**. La expresión *selector* debe ser un tipo ordinal (por ejemplo, *int*, *char*, *bool* pero no *float* o *string*).

Cada **etiqueta** es un valor único, constante y tiene un valor diferente de los otros.

Si el valor de la expresión selector es igual a una de las etiquetas *case* –por ejemplo *etiqueta2*– entonces la ejecución comenzará con la primera sentencia que proceda a ese *case*, y continuará hasta que se encuentra una sentencia *break* (o hasta que se encuentra el final de la estructura *switch*)

Instrucción *default*

Si el valor del selector no está listado en ninguna etiqueta *case*, no se ejecutará ninguna de las opciones a menos que se especifique una acción por defecto mediante la palabra reservada *default*. La omisión de una etiqueta *default* puede crear un error lógico difícil de prever. Aunque la etiqueta *default* es opcional, se recomienda su uso a menos que se esté absolutamente seguro de que todos los posibles valores del selector estén incluidos en las etiquetas *case*.

Instrucción *break*

Cuando la computadora ejecuta las sentencias siguientes a una etiqueta *case*, continúa hasta que se alcanza una sentencia *break*. Si la computadora encuentra una sentencia *break*, termina la estructura *switch*.

Si se omiten las sentencias *break*, después de ejecutar el código de *case*, la computadora ejecutará el código correspondiente al siguiente *case*.

¡PRECAUCION!

Si se olvida la instrucción *break* en una estructura *switch*, el compilador no emitirá un mensaje de error, ya que la estructura *switch* es “sintácticamente correcta”, pero no obtendrá el resultado esperado.

Dentro de la sentencia *switch*, está permitido tener varias expresiones *case* para ejecutar a un mismo segmento de código cuando se cumpla alguna de ellas.

Por ejemplo, se puede escribir:

```
switch(caracter){
    case 'a': case 'A': case 'e': case 'E': case 'i': case 'I':
    case 'o': case 'O': case 'u': case 'U':
        cout<<caracter<<" es una vocal"<<endl;
    break;
    case '1': case '2': case '3': case '4': case '5': case '6':
    case '7': case '8': case '9': case '0':
        cout<<caracter<<" es un digito"<<endl;
    break;
    default:
        cout<<caracter<<" no es ni vocal ni digito"<<endl;
}
```

Uso de la estructura switch para la construcción de menús.

La estructura *if-else* es más versátil que la estructura *switch* y se pueden utilizar sentencias *if-else* anidadas en cualquier parte en la que se utiliza *switch*. Sin embargo, normalmente, la estructura *switch* es más clara. Por ejemplo, la estructura *switch* es idónea para implementar menús.

Cualquiera de los menús de un restaurante representa una lista de alternativas para que un cliente seleccione entre las diferentes opciones.

Un menú en un programa de computadora hace la misma función: presentar una lista de alternativas en la pantalla para que el usuario elija una de ellas.

III. MATERIALES Y EQUIPO

No.	Requerimiento	Cantidad
1	Compilador de C++	1
2	Practica 6: Estructura selectiva (switch)	1
3	Memoria USB	1

IV. PROCEDIMIENTO

1. Cree una carpeta llamada **PALpractica6deCARNET** en su PC para que guarde sus archivos .cpp del procedimiento de la guía y también de los ejercicios solicitados en el análisis de resultados. Recuerde reemplazar CARNET por su respectivo número de carnet.

2. Ahora desarrolle cada uno de los programas en C++, teniendo cuidado de aplicar las reglas siguientes en cada código fuente:

- Cuide el uso de mayúsculas y minúsculas al redactar palabras reservadas y los nombres de variables y/o funciones, **ya que lenguaje C++ es sensible a mayúsculas y minúsculas.**
- Realice la compilación respectiva, con el fin de generar el archivo .exe de aplicación.
- Desarrolle varias pruebas de ejecución, ingresando diversos datos en cada prueba, para construir una idea general del funcionamiento del programa
- Cuando encuentre una nota de **IMPORTANTE**, al final de un código, realice sin falta las pruebas solicitadas ahí, para deducir las respuestas apropiadas

1. Código cpp del Programa: PAL_Ejemplo1

En algunos países, los sistemas de evaluación académica utilizan letras en lugar de números para evaluar el rendimiento de los alumnos.

El siguiente código, ejemplifica el uso de la estructura switch: Se pide el ingreso de una calificación (letra, entre la A a la F), para así mostrar una evaluación de dicha calificación.

```
#include<iostream>
using namespace std;
#include<windows.h>
#include<conio.h>

const char en=(char) 164;

main(){
    //para acceder al cursor de la ventana de aplicacion
    HANDLE cur;
    cur = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD p;

    char nota; //Nota a evaluar

    p.Y=4; p.X=24; SetConsoleCursorPosition(cur,p);
    cout<<"Evaluacion de resultado de examen";

    p.Y=7; p.X=6; SetConsoleCursorPosition(cur,p);
    cout<<"Ingrese su nota (letra A hasta F ) del examen final: ";

    p.Y=9; p.X=6; SetConsoleCursorPosition(cur,p);
    cout<<"RESULTADO: ";

    p.Y=7; p.X=60; SetConsoleCursorPosition(cur,p);
    nota=getche();

    p.Y=9; p.X=17; SetConsoleCursorPosition(cur,p);
    switch(nota){
        case 'A': case 'a':
            cout<<"Excelente, aprobado con honores";
            break;
        case 'B': case 'b':
            cout<<"Aprobado de forma sobresaliente";
            break;
        case 'C': case 'c':
            cout<<"Aprobado con buen desempe"<<en<<"o";
            break;
        case 'D': case 'd':
            cout<<"Aprobado de manera regular";
            break;
        case 'E': case 'e':
```

```

    cout<<"Aprobado, pero se sugiere mejorar";
    break;
case 'F': case 'f':
    cout<<"Reprobado, con rendimiento pobre";
    break;
default:
    cout<<"** ERROR, Nota ingresada es incorrecta **";
} //fin switch nota
getch();
}

```

IMPORTANTE:

- Pruebe el programa ingresando una letra (entre la A a la F) y evalúe el resultado devuelto. Luego ingrese cualquier otra letra, por ejemplo "T" y verifique la salida obtenida. ASEGURESE DE ENTENDER POR QUÉ SE OBTIENE ESE RESULTADO.
- **Funciones getch() y getche():** Estas 3 funciones se especializan en capturar UN SOLO CARÁCTER y constituyen la mejor forma de leer una opción de menú (la función estándar cin se queda esperando un "enter" razón por la cual no resulta útil cuando se quiere leer un dato de tipo char).

La diferencia entre estas funciones, es que `getchar` devuelve el carácter leído a la pantalla y `getch` no devuelve el carácter a la pantalla.

2. Código cpp del Programa: PAL_Ejemplo2

Crear una calculadora de funciones trigonométricas.

Dado un ángulo ingresado por el usuario (en grados) se debe calcular a una de las 3 funciones trigonométricas básicas.

```

#include<iostream>
#include<stdlib.h>
#include<conio.h>
#include<math.h>
#include<windows.h>

using namespace std;

main(){
    const float PI = 3.14159265;
    char g = (char) 248; //simbolo grado
    //para acceder al cursor de la ventana de aplicacion
    HANDLE cur;
    cur = GetStdHandle(STD_OUTPUT_HANDLE);
}

```

```

COORD p;

//variables de entrada
float angulo=0;
char opcion=' ';//Almacenará opción seleccionada por el usuario
//variable de proceso, guarda la conversión grados-radianes
float anguloRadianes=0;
float resul=0;//Variable de salida
cout<<"\t* CALCULADORA FUNCIONES TRIGONOMETRICAS BASICAS *\n";
cout<<"\t*****\n";
cout<<"\t* Lista de funciones trigonometricas *"<<endl;
cout<<"\t* 1- Seno *"<<endl;
cout<<"\t* 2- Coseno *"<<endl;
cout<<"\t* 3- Tangente *"<<endl;
cout<<" \t*****"<<endl;

p.X= 5; p.Y=10; SetConsoleCursorPosition(cur,p);
cout<<"Selecciona el numero correspondiente a la funcion deseada: ";

p.X= 5; p.Y=16; SetConsoleCursorPosition(cur,p);
cout<<"RESULTADO: ";

p.X= 65; p.Y=10; SetConsoleCursorPosition(cur,p);
opcion=getche();//Capturamos la seleccion del usuario

switch(opcion){
case '1': case '2': case '3':
p.X= 5; p.Y=13; SetConsoleCursorPosition(cur,p);
cout<<"Ingresa el valor del angulo (en grados sexagesimales): ";
cin>>angulo;

anguloRadianes=angulo*(PI/180); //Convierte angulo a radianes

p.X= 16; p.Y=16; SetConsoleCursorPosition(cur,p);

switch(opcion){
case '1':
resul=sin(anguloRadianes);
cout<<"Seno de "<<angulo<<g<<" es: "<<resul;
break;
case '2':
resul=cos(anguloRadianes);
cout<<"Coseno de "<<angulo<<g<<" es: "<<resul;
break;

```

```

    case '3':
        resul=tan(anguloRadianes);
        cout<<"Tangente de "<<angulo<<g<<" es: "<<resul;
    }//fin switch

    break;

default:
    p.X= 16; p.Y=16; SetConsoleCursorPosition(cur,p);
    cout<<"\a\aopcion no valida"; // \a=sonido timbre
} //fin switch

cout<<"\n\n\n";
system("pause");
} //fin main

```

IMPORTANTE:

- Revise cuidadosamente cada línea del ejemplo anterior, y responda ¿Por qué se declara la variable “opcion” del tipo char si las opciones están como números enteros?
- Complete la siguiente tabla de resultados (y agregue resultados de 4 pruebas mas):

Prueba	Entradas		Salida	Prueba	Entradas		Salida
#	opcion	angulo	resul	#	opcion	angulo	resul
1	1	90	1	5			
2	3	90		6			
3	2	60		7			
4	2	45		8			

¿Cuál fue el resultado de la prueba 7 y 8? Asegúrese de entender el porqué de este resultado.

3. Nombre de código fuente: PAL_Ejemplo3
El siguiente programa determina si un carácter ingresado por el usuario es vocal, dígito o ninguno de los anteriores.

```

#include<iostream>
#include<stdlib.h>
#include<conio.h>

using namespace std;

```

```

main(){
    char caracter;
    cout<<"\n\tIngresa un caracter: ";
    caracter=getche();

    switch(caracter){
        case 'a': case 'A': case 'e': case 'E':
        case 'i': case 'I': case 'o': case 'O':
        case 'u': case 'U':
            cout<<"\n\t El caracter "<<caracter<<" es una vocal"<<endl;
            break;
        case '1': case '2': case '3': case '4': case '5': case '6':
        case '7': case '8': case '9': case '0':
            cout<<"\n\t El caracter "<<caracter<<" es un digito"<<endl;
            break;
        default:
            cout<<"\n\t caracter "<<caracter<<" no es ni vocal ni digito"<<endl;
    }
    system("PAUSE");
}

```

4. Nombre de código fuente: PAL_Ejemplo4

El Estadio Cuscatlán es uno de los estadios más importantes de Centroamérica. Este estadio tiene diversos sectores. El costo de la entrada a los eventos futbolísticos del estadio se asignan en virtud de los sectores del estadio mediante la siguiente tabla:

<i>Sector</i>	<i>Costo de la entrada</i>
Sol general	\$3
Sol preferente	\$5
Sombra	\$8
Tribuna	\$15
Platea	\$20

Se pide construir un programa que permita seleccionar un sector del estadio, ingresar la cantidad de entradas solicitadas y calcular el total a pagar por las entradas.

```

#include<iostream>
using namespace std;

```

```

#include<stdlib.h>
#include<conio.h>

main(){
    int cantidad=0;//Almacena la cantidad de entradas compradas

//Almacena la opcion de menu seleccionada por el usuario
    char opcion=' ';
    float precio=0, total=0;

    cout<<"\n\t*****" <<endl;
    cout<<"\t*      BIENVENIDO AL ESTADIO CUSCATLAN      *" <<endl;
    cout<<"\t*****" <<endl;
    cout<<"\t* Sectores del estadio                               *" <<endl;
    cout<<"\t*   A- Sol general                                       *" <<endl;
    cout<<"\t*   B- Sol preferente                                    *" <<endl;
    cout<<"\t*   C- Sombra                                           *" <<endl;
    cout<<"\t*   D- Tribuna                                           *" <<endl;
    cout<<"\t*   E- Platea                                           *" <<endl;
    cout<<"\t*****" <<endl;
    cout<<"\n\t Selecciona la letra del sector del estadio : ";
    opcion=getche(); //Capturamos la seleccion del usuario
    cout<<"\n\t Ingresa la cantidad de entradas requeridas: ";
    cin>>cantidad;
    if(cantidad<1){ //Cantidad de entradas no puede ser menor de 1
    cout<<"\n\t\a ERROR: Cantidad de entradas debe ser mayor que 1" <<endl;
        system("PAUSE");
        return 0;//si pone menos de 1 el programa "se cierra"
    }
    system("cls");
    switch(opcion){
        case 'A': case 'a':
            cout<<"\n\t Sector seleccionado: Sol general" <<endl;
            precio=3;
            break;
        case 'B': case 'b':

```

```

        cout<<"\n\t Sector seleccionado: Sol preferente"<<endl;
        precio=5;
        break;
    case 'C': case 'c':
        cout<<"\n\t Sector seleccionado: Sombra"<<endl;
        precio=8;
        break;
    case 'D': case 'd':
        cout<<"\n\t Sector seleccionado: Tribuna"<<endl;
        precio=15;
        break;
    case 'E': case 'e':
        cout<<"\n\t Sector seleccionado: Platea"<<endl;
        precio=20;
        break;
    default:
        cout<<"\n\t\a ERROR: El sector seleccionado no existe"<<endl;
        system("PAUSE");
        return 0;//Cerramos el programa
    }
    total=precio*cantidad;//Calculando total a pagar
    cout<<"\t Precio unitario: $"<<precio<<endl;
    cout<<"\t Cantidad de entradas:"<<cantidad<<endl;
    cout<<"\t TOTAL A PAGAR: $"<<total<<endl;
    cout<<"\n\n\t GRACIAS POR VISITARNOS!"<<endl;
    system("PAUSE");
}

```

IMPORTANTE:

- Observe que las lecturas de datos están “validadas”, esto significa que en caso que el usuario ingrese valores incorrectos se genera un mensaje que indica al usuario cuál es el error cometido.
- Pruebe lo anterior ingresando un número de sector inexistente (por ejemplo 10) o una cantidad de entradas incorrecta (por ejemplo -5).

V. ANÁLISIS DE RESULTADOS

- Elabore en C++ a las soluciones de cada uno de los problemas descritos a continuación.
- Cada código fuente no debe utilizar ninguna estructura de decisión (`if`), solamente estructuras (`switch`). Se descontara 50% si no se cumple esta restricción.

PROBLEMA 1:

Modifique el ejemplo PAL_Ejemplo2 del procedimiento, de tal forma que usuario pueda elegir entre ingresar el ángulo solicitado en grados sexagesimales o en radianes.

Ademas, extienda el menú de operaciones para que permita calcular el resto de funciones trigonométricas (*cotangente*, *secante* y *cosecante*).

PROBLEMA 2:

Solicite al usuario el diámetro y la altura (ambas medidas en centímetros) de un cilindro, para calcular su área total exterior. Luego solicite a usuario en cual tipo de medidas desea ver el resultado. Las alternativas disponibles para presentar el resultado son:

- a) milímetros cuadrados b) centímetros cuadrados c) pulgadas cuadradas

PROBLEMA 3:

Un municipio desea cobrar por el paso de los automotores por su nuevo Puente. Para ello ha creado la siguiente tabla de peajes según el tipo de automotor.

Automotor	Peaje
Vehículo particular	\$0.75
Autobús	\$1.50
Microbús	\$1.25
Motocicleta	\$0.25
Moto taxi	\$0.50
Otros	\$3.00

Se pide escribir un programa en C++ que muestre el peaje a pagar según el tipo de automotor.

PROBLEMA 4:

Modifique el código fuente del Ejemplo 3 del procedimiento de esta practica, de tal forma que la aplicación evalúe a la tecla ingresada (que puede estar en minúscula o mayúscula) y luego indique si es:

- A. Letra vocal B. Letra consonante C. Dígito (0, 1, 9)

PROBLEMA 5:

Elabore un conversor de medidas de longitud. El usuario podrá elegir entre las siguientes unidades de medida:

- a) Metros b) Pies c) Centímetros d) Pulgadas

PROBLEMA 6:

La empresa “PRINTFACIL S.A de C.V” se dedica a vender impresores con sistema de tinta continuo a un precio unitario de \$78 (sin IVA).

Se le pide que escriba un programa en C++ que permita ingresar la cantidad de impresores a comprar y seleccionar la forma de pago para luego generar el detalle del pago a realizar.

Según la forma de pago, la empresa realiza los siguientes descuentos:

Forma de pago	Descuento
Efectivo	10%
Tarjeta de crédito	5%
Vale de regalo	15%

Finalmente, el programa debe mostrar:

- La cantidad de impresoras a comprar
- El precio unitario de la impresora (con IVA).
- El total sin descuento
- La forma de pago
- El descuento realizado
- El total a pagar

Considere que todos los descuentos se realizan sobre el precio CON IVA.

PROBLEMA 7:

Brinde su ayuda a un estudiante de electrónica, el cual busca calcular a una resistencia equivalente que reemplace a 3 dispositivos iniciales, los pueden estar conectados según lo indique el estudiante (que puede ser en serie, en paralelo o mezclas de ambas maneras, como muestra la *Figura 1*. Todas las resistencias se miden en kilo-ohmios.

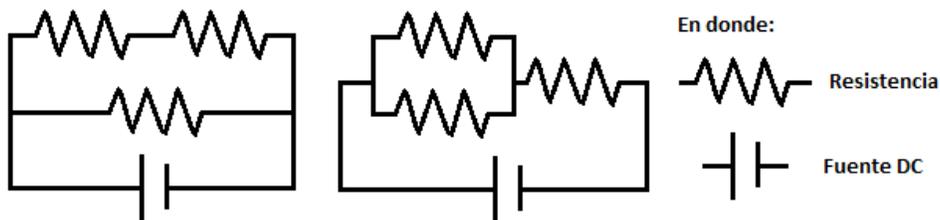


Figura 1: Tipos de conexiones mixtas entre 3 resistencias

Guía de Laboratorio No. 6

RÚBRICA DE EVALUACIÓN

Actividad a evaluar: ANÁLISIS DE RESULTADOS

Formar grupos entre 3 a 5 estudiantes, llenar esta hoja de evaluación y entregarla a su docente.

Luego, su instructor seleccionará 3 problemas de la DISCUSION DE RESULTADOS, para ser resueltos apropiadamente por el grupo.

Lista de Integrantes:

CARNET 1	CARNET 2	CARNET 3	CARNET 4	CARNET 5

Problemas a resolver:

Criterio a evaluar	¿Prob 1?	¿Prob 2?	¿Prob 3?	PROMEDIO	Puntaje
(15%) Define las variables de entrada y salida esperadas					
(20%) Se redactan correctamente las sentencias (switch). No utiliza ninguna estructura (if)					
(25%) Código fuente se compila y obtiene los resultados solicitados					
(20%) Se cumplen cada una de las restricciones descritas en el problema					
(20%) Documenta apropiadamente cada ejercicio solicitado					
Nota:					