Ingeniería de Software I
Pruebas de Software

Objetivo

Verificar el uso de las estructuras de control selectivas if y switch para la solución de un problema mediante el uso exclusivo de if y comparar la estructuración de código para el caso de comparación contra tipos enteros utilizando switch.

Aprender a realizar pruebas de software como caja negra y caja blanca.

Descripción

La tarea trata sobre la verificación de varias fechas hardcodeadas, para determinar si son válidas o no. No se requerirá capturar del usuario información alguna.

```
...
if (esFechaValida(1979,SEPTIEMBRE,12)){
    ...
}
...
```

Años

Cualquier año sea válido, esto es, en el rango de los enteros con ó sin signo.

Meses

El número correspondiente a cada mes a validar es como sigue:

Mes	Número de mes
Enero	1
Febrero	2
Marzo	3
Abril	4
Mayo	5
Junio	6
Julio	7
Agosto	8
Septiembre	9
Octubre	10
Noviembre	11
Diciembre	12

Días

Los días válidos para cada mes son como sigue:

Mes	Días
Enero	Del 1 al 31
Febrero	Del 1 al 28 para todos los años 29 para los años bisiestos
Marzo	Del 1 al 31
Abril	Del 1 al 30
Mayo	Del 1 al 31
Junio	Del 1 al 30
Julio	Del 1 al 31
Agosto	Del 1 al 31
Septiembre	Del 1 al 30
Octubre	Del 1 al 31
Noviembre	Del 1 al 30
Diciembre	Del 1 al 31

Años bisiestos

Un año bisiesto se presenta cada 4 años, contados a partir del año cero hacia adelante y hacia atrás, esto es:

-4, 0, 4, 8, 12…etc.

…son años bisiestos.

Hay una excepción para los años bisiestos, esto es cada 100 años:

100, 200, 300, 500, 600, 700…etc.

…no son años bisiestos.

Cómo se habrá notado en la serie anterior, hay una excepción a la excepción, esto es que cada 400 años:

400, 800, 1200…etc.

…sí son años bisiestos.

La tarea consiste de 3 variantes del mismo programa (3 códigos fuente que ante el usuario hacen lo mismo pero están programados diferentes al interior) y una optativa.

Ejemplos de validaciones son los siguientes:

```
VERIFICADOR DE FECHA v1.0

Probando fecha aaaa/mm/dd: 1979/9/12
Fecha válida!
Presione entrar para continuar . . .
Probando fecha aaaa/mm/dd: 2011/1/0
Día no válido!
Presione entrar para continuar . . .
Probando fecha aaaa/mm/dd: 2011/13/1
Mes no válido!
Presione entrar para continuar . . .
Probando fecha aaaa/mm/dd: 2011/-1/12
Mes no válido!
Presione entrar para continuar . . .
Probando fecha aaaa/mm/dd: 2011/4/31
Día no válido!
Presione entrar para continuar . . .
```

Para el caso de día y mes no válidos es admisible:

```
Probando fecha aaaa/mm/dd: 2011/0/0
Día no válido!
```

O bien

```
Probando fecha aaaa/mm/dd: 2011/0/0
Mes no válido!
```

Para lo anterior, generar una función de validación para la fecha y además, programar en la función main varias pruebas de caja negra a criterio propio para revisar la correcta ejecución de la función.

Requerimientos Generales y Funcionales

1. Entregar archivos *fuente* para tres aplicaciones de consola que cumplan con la citada descripción y los siguientes requerimientos.
2. Colocar los archivos en una carpeta distinta, nombradas: “variante1”, “variante2”, “variante3” y “variante4”, acorde a requerimientos particulares en el apartado siguiente.
3. Imprimir en consola el título de la aplicación.
4. Realizar la verificación de la fecha de forma tal que se imprima una y sola una de las siguientes salidas según corresponda:
   a) “Fecha válida”
   b) “Mes no válido”
   c) “Día no válido”

   …ninguna otra impresión a la consola, más que las citadas, sea impresa.

5. Después de imprimir el resultado del punto anterior, mostrar el mensaje “Presione entrar para continuar . . .” y para continuar esperar a que se presione dicha tecla.

Requerimientos Particulares

Para cada una de las siguientes variantes la implementación consista de:

6. Variante 1: el código fuente para validación de la fecha use exclusivamente la selectiva if sin anidamientos y sin usar operador de condición ni banderas
7. Variante 2: el código fuente para validación de la fecha use exclusivamente la selectiva if con anidamientos y sin usar operador de condición ni banderas
8. Variante 3: el código fuente para validación de la fecha use una combinación de la selectiva if con la switch anidadas, usando switch para los casos de los meses y sin usar operador de condición ni banderas.

Requerimientos No Funcionales

9. En caso de lenguaje C++/Java, para las banderas usar en lugar de variables tipo int el tipo bool/boolean.
10. El código fuente de cada archivo cumpla con la siguiente estructura general de programa:
    Declaración y definición de constantes
    Declaración de variables globales
    Prototipos de procedimientos/funciones
    Definición de subrutinas
    …donde el código fuente de la función *main* cumpla con la siguiente estructura de programa:
    Declaración de variables para cómputo
    Título de la aplicación
    Cómputo de casos de prueba e impresión de resultados
11. Implementar una función llamada esFechaValida(), la cual reciba 3 parámetros tipo entero “anio”, “mes” y “dia”, en dicho orden y devuelva el valor 1 solo si es una fecha válida ó 0 (cero) en caso contrario; en caso de C++/Java, devolver un tipo bool o boolean.
12. Declarar dentro de esFechaValida() las variables (que sean necesarias) para almacenamiento de resultados del cómputo de datos de entrada
13. Declarar una variable global nombrada “codigoErrorFecha” que sirva para almacenar el resultado de la validación, la cual durante la ejecución de la función esFechaValida() se establezca en 0 cuando sea el caso de “Fecha válida”, -1 en el caso de “Mes no válido” y -2 en el caso de “Día no válido”
14. Implementar una función llamada dameCodigoErrorFecha() que devuelva el contenido de la variable global “codigoErrorFecha”

15. Las funciones esFechaValida(), dameCodigoErrorFecha() y la variable global mencionada anteriormente, se encuentren dentro de un archivo de librería llamado "Fecha.h"
16. Programar en la función *main* pruebas de caja negra "hardcodeados" a criterio propio, esto es, llamar a la función de validación (con parámetros para año, mes y día), pasándole argumentos de combinaciones de valores correctos e incorrectos, intentando corroborar que la función valida correctamente o falla en la etapa de pruebas.
17. Realizar la verificación de la fecha de forma tal que se imprima una y sola una de las salidas descritas en el requerimiento 4.
18. Llamar a la función de validación al menos 10 veces, mostrando los resultados de las pruebas según el requerimiento 4.
19. Evitar imprimir a consola dentro de la librería "Fecha.h".
20. Todo valor izquierdo debe ser utilizado en el programa.
21. Para los requerimientos 6, 7 y 8 se permite tener en la función de validación varias ocurrencias de la sentencia return, sin embargo es deseable evitarlo de ser posible.
22. Para imprimir el mensaje a consola según el requerimiento 4, utilizar la estructura de control switch.
23. El programa principal (main) sea el mismo código fuente en todas las variantes.
24. Crear una subcarpeta variante4 con un archivo *fuente* para una aplicación de consola que cumpla con los siguientes requerimientos.
25. Para determinar cada una de las salidas descritas en el requerimiento 4 se utilicen 3 banderas fechaValida, mesValido, diaValido y en la subrutina esFechaValida() aplicar un y solo un return.
26. A toda variable le sea asignado el valor que le corresponda una vez y solo una vez.

Requerimientos para Puntos Extras

27. En el caso de entregar en lenguajes C o C++, declarar la variable global "codigoErrorFecha" de modo que esta sea visible solo dentro del archivo donde se declare.
28. Entregar en el mismo comprimido carpetas diferentes para cada lenguaje:
    a) Una nombrada "C" con los códigos fuente en C (ANSI);
    b) Una nombrada "C++" con los códigos fuente en C++;
    c) Una nombrada "Java" con los códigos fuente en Java;
    d) Una nombrada "C#" con los códigos fuente en C#;
    e) Una nombrada "English" con los códigos fuente escritos en su totalidad en inglés (solo para el lenguaje de su preferencia).

    …las primeras opciones se redacten absolutamente en español y la última absolutamente en inglés. Si incluye 2 carpetas con lenguajes OO, uno de ellos puede redactarse en inglés y sin necesidad de entregar la versión en español para ese lenguaje.

Criterios de Evaluación

- Los establecidos en las "Reglas de Operación y Evaluación" del curso.
- Cumplir con la fecha límite de entrega citada en el Excel de Actividades.

- Calificación en base a cobertura de requerimientos y fecha de entrega.
- Cumplir con Requerimientos de Valor Agregado en Código Fuente (hasta el req. “FF”).
- Entrega en un lenguaje: Java o C#.
- Es indispensable la entrega de un programa cuyo código fuente sea completamente en idioma Español (a excepción de lo correspondiente a la API del lenguaje).