



## Laboratorio Clase 4

- Practico sobre Estructuras de SELECCIÓN

IF – ELSE / SWITCH



## Laboratorio Clase 4

Página 222 Bronson.

*Escriba un programa en C++ para calcular la raíz cuadrada y el reciproco de un numero introducido por el usuario. Antes de calcular la raíz cuadrada deberá verificar que el numero NO sea negativo; y antes de calcular el reciproco deberá verificar que el numero NO sea 0.*

*BONUS PACK: La impresión de la raíz cuadrada y del reciproco debe tener una precisión de 3 lugares decimales.*

### Primer paso: Análisis básico:

Encontrar palabras como: calcular, imprimir, determinar, encontrar o comparar.

En este caso la salida debe ser: “calcular la raíz cuadrada y el reciproco”.  
Por lo tanto hay dos salidas.



## Laboratorio Clase 4

Una vez obtenida la salida, se debe analizar las entradas necesarias para poder producir esa salida.

Es importante distinguir los elementos necesarios para el calculo de la salida pero no lo valores que tienen ellos.

El elemento de entrada de este programa **es el numero ingresado por el usuario.**

Podemos ver que el algoritmo para la solución esta dado por la formula

Para el caso de la raíz cuadrada:  $\text{sqrt}(\text{numero})$

Para el caso del reciproco:  $1/(\text{numero})$

### **Paso 2: Desarrollar una solución**

El algoritmo nos da la formula, ahora debe refinarse paso a paso en detalle como han de combinarse las entradas para producir la salida.



## Laboratorio Clase 4

### Pseudo código:

*Muestro un indicador de comandos para que el usuario ingrese un valor  
Asignar el valor ingresado a una variable*

*Pregunto si el numero es positivo? Calculo la raíz cuadrada e imprimo  
Pregunto si el numero es 0? Calculo el reciproco e imprimo en función al  
resultado*

Ahora corresponde hacer una prueba manual en lápiz y papel del algoritmo para verificar que cumpla el objetivo.

Esto nos va a permitir contrastar en la etapa final de prueba del programa si el resultado es el mismo. EJ: numero = 25  
La raíz cuadrada es 5, y el reciproco 0.04

### Paso 3: Codificar la solución

Se necesita declarar variables para cada entrada y para la salida.



```
#include <cmath>
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    double numero=0;
    cout<<"Este programa calcula la raiz cuadrada y\n"
        <<"el reciproco (1/numero) de un numero \n"
        <<"\n Por favor introduzca un numero: ";
    cin>>numero;
    if (numero < 0.0)
        cout<<"La raiz cuadrada de un numero negativo no existe.\n";
    else
        {cout<<fixed<<setprecision(3);
         cout<<"La raiz cuadrada de "<<numero<<"es"<<sqrt(numero)<<endl;}
    if (numero == 0.0)
        cout<<"El reciproco de 0 no existe.\n";
    else
        cout<<"El reciproco de "<<numero<<"es"<<(1/numero)<<endl;
    system("pause");
    return 0;
}
```

### Codificación:

Indicador de comandos



### *Ejecución en DEV C++*

Formato de salida

FALTAN  
COMENTARIOS!!!!!!



## Laboratorio Clase 4

### **Paso 4: Prueba y corrección**

El objetivo de esta etapa es verificar que funciona en forma correcta y en realidad satisface los requerimientos.

Aquí verificamos la respuesta del programa, al calculo realizado en forma manual.

Debido a que este programa solo realiza dos cálculos, y que la salida producida por la ejecución de prueba coincide con el calculo manual se tiene un buen grado de confianza de que puede ser usado para el calculo de otros valores de entrada



## Laboratorio Clase 4

### Versión EJERCICIO 2<sup>a</sup> y 2<sup>c</sup> - página 228 Bronson.

2. a. Escriba un programa para desplegar los dos indicadores siguientes:

Introduzca un mes (use 1 para Ene, etc.):

Introduzca un dia del mes:

Haga que su programa acepte y almacene un número en la variable `mes` en respuesta al primer indicador, y acepte y almacene un número en la variable `dia` en respuesta al segundo indicador. Si el mes introducido no está entre 1 y 12 inclusive, imprima un mensaje informando al usuario que se ha introducido un mes inválido. Si el día introducido no está entre 1 y 31, imprima un mensaje informando al usuario que se ha introducido un día inválido.

- c. En un año que no es bisiesto, febrero tiene 28 días, los meses de enero, marzo, mayo, julio, agosto, octubre y diciembre tienen 31 días y todos los demás meses tienen 30 días. Usando esta información, modifique el programa escrito en el ejercicio 2a para desplegar un mensaje cuando se introduzca un día inválido para un mes introducido por un usuario. Para este programa ignore los años bisiestos.

En ambos casos, si los valores son válidos imprimir en pantalla la fecha



## Laboratorio Clase 4

### EJERCICIO 4 - página 229 Bronson.

Todos los años que se dividen **exactamente** entre 400 o que son divisibles **exactamente** entre 4 y no son divisibles **exactamente** entre 100 son años bisiestos. Por ejemplo en vista que 1600 es divisible exactamente entre 400, el año 1600 fue un año bisiesto. Del mismo modo, en vista que 1988 es divisible exactamente por 4 y no es divisible exactamente por 100, el año 1988 también fue un año bisiesto.

Usando esta información escriba un programa en C++, que acepte el año como una entrada del usuario, determine si el año es un año bisiesto y despliegue un mensaje apropiado que le indique al usuario si el año introducido es o no bisiesto.

Ayuda: revise las operaciones aritméticas vistas en clase.