**Tarea\_02**

Edward Nicolas Duarte Valencia

**Diseño de Sistemas con Procesadores**

Departamento de Ingeniería Electrónica

Pontificia Universidad Javeriana

Realización: 08/03/2021 - Entrega: 08/03/2021

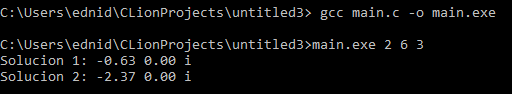
**Código:**

*/\* Ruler 1 2 3 4 5 6 7 \*/  
/\* Tarea 2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
/\* \*/  
/\* ┌────┐ ┌────┐ ECUACION CUADRATICA \*/  
/\* └┐ ┌┘ └┐╔══╧═╗ \*/  
/\* │ │ │╚╗ ╔╝ DEVELOPED BY: Ing. Edward Duarte \*/  
/\* │ │ │ ║ ║ edwardduarte@javeriana.edu.co \*/  
/\* │ │ │ ║ ║ \*/  
/\* │╔═╧══╗│ ║ ║ Bucaramanga, March 08th, 2021. \*/  
/\* │╚╗ ╔╝┘ ║ ║ \*/  
/\* └┐║ ╚╗ ╔╝ ║ Copyright (C) Electronics Engineering Program \*/  
/\* └╚╗ ╚═╝ ╔╝ School of Engineering \*/  
/\* └╚╗ ╔╝ Pontificia Universidad Javeriana \*/  
/\* ╚═════╝ Bogota - Colombia - South America \*/  
/\* \*/  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* bench.h \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
/\* \*/  
/\* Purpose: Hallar las raices de una función cuadratica \*/  
/\* \*/  
/\* Origin: Ingeniero Juan Carlos Giraldo \*/  
/\* \*/  
/\* Notes: Este codigo está diseñado para ser corrido en cmd \*/  
/\* \*/  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* HEADERS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
#include* <stdio.h>  
*#include* <math.h>  
*#include* <aclapi.h>  
*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Estructuras \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*typedef struct {  
 double real;  
 double imag;  
} *COMPLEX\_T*;  
  
*/\*FN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\* prototype: int solve\_Quadratic\_Ecuation(COMPLEX\_T \*root\_1,COMPLEX\_T \*root\_2,  
\* double a,double b,double c)  
\*  
\* Return: 1  
\*  
\* Purpose: Hallar las raices de una ecuación cuadratica con entrada por param.  
\*  
\* Plan:  
\* 1) Definir los parametros de entrada de la función, con 3 datos de  
\* tipo double y dos apuntadores a estructuras de tipo COMPLEX\_T  
\* 2) Crear una variable con el discriminante, el cual permitira la selección  
\* entre los distintos casos posibles  
\* 3) Condicionales que establecen el tipo de respuesta de las raíces  
\* 3.1) Caso de dos raices diistintas con valores reales.  
\* 3.2) Caso de 1 raiz con valor real.  
\* 3.3) Caso de dos raices diistintas con valores imaginarios.  
\*  
\* Register of Revisions:  
\*  
\* DATE RESPONSIBLE COMMENT  
\* -----------------------------------------------------------------------  
\* marzo 08/21 J.C.Giraldo Initial implementation  
\* marzo 08/21 Edward Duarte Function developer  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCION\_EC.CUADRATICA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
/\*----------------------------------1)---------------------------------------\*/*int Solve\_Quadratic\_Equation(  
 double *a*,  
 double *b*,  
 double *c*,  
 *COMPLEX\_T* \**root\_1*,  
 *COMPLEX\_T* \**root\_2* )  
{  
*/\*----------------------------------2)---------------------------------------\*/* double discriminante;  
 discriminante = *b* \* *b* - 4 \* *a* \* *c*;  
*/\*----------------------------------3)---------------------------------------\*/  
  
/\*----------------------------------3.1)-------------------------------------\*/* if(discriminante > 0) {  
 *root\_1*->real = (-*b* + sqrt(discriminante)) / (2 \* *a*);  
 *root\_1*->imag = 0;  
 *root\_2*->real = (-*b* - sqrt(discriminante)) / (2 \* *a*);  
 *root\_2*->imag = 0;  
 }  
*/\*----------------------------------3.2)-------------------------------------\*/* else if (discriminante == 0) {  
 *root\_1*->real = -*b* / (2 \* *a*);  
 *root\_1*->imag = 0;  
 *root\_2*->real = -*b* / (2 \* *a*);  
 *root\_2*->imag = 0;  
 }  
*/\*----------------------------------3.3)-------------------------------------\*/* else {  
 *root\_1*->real = -*b* / (2 \* *a*);  
 *root\_1*->imag = sqrt(-discriminante) / (2 \* *a*);  
 *root\_2*->real = -*b* / (2 \* *a*);  
 *root\_2*->imag = -sqrt(-discriminante) / (2 \* *a*);  
 }  
} */\*solve\_Quadratic\_Ecuation\*/  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCION\_PRINCIPAL \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*int main( int *argc*, char \**argv*[])  
{  
 double a = atof( *argv*[1] );  
 double b = atof( *argv*[2] );  
 double c = atof( *argv*[3] );  
  
 *COMPLEX\_T* root\_1, root\_2;  
 Solve\_Quadratic\_Equation( a, b, c, &root\_1, &root\_2 );  
 fprintf( **stdout**, "Solucion 1: %.2f %.2f i\n", root\_1.real, root\_1.imag );  
 fprintf( **stdout**, "Solucion 2: %.2f %.2f i\n", root\_2.real, root\_2.imag );  
  
 return 0;  
  
}*/\*main\*/*

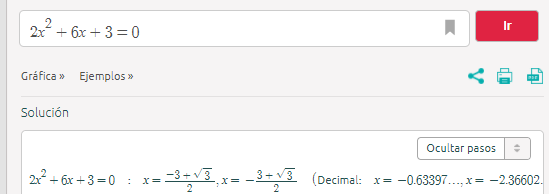
**Comprobación:**

**1 caso: 2x^2+6x+3=0 (Raíces reales y diferentes)**

**Solución en cmd:**

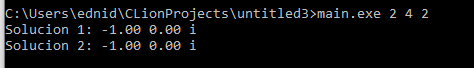
****

**Solución en symbolab:**

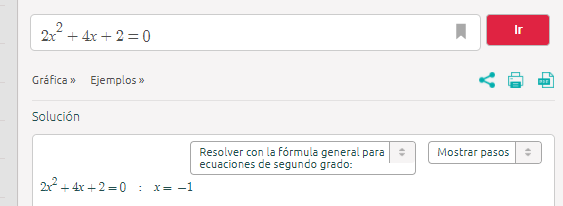
****

**2 caso: 2x^2+4x+2=0 (raíces iguales)**

**Solución en cmd:**

****

**Solución en symbolab:**

****

**3 caso: 1x^2+2x+3=0 (Raíces con parte imaginaria)**

**Solución en cmd:**

****

**Solución en symbolab:**

****

**4 caso: 2x^2+100x+3=0 (Raíces con parte real)**

**Solución en cmd:**

****

**Solución en symbolab:**

****

****

**6 caso: 9999x^2+12345x+9876546654=0**

**Solución en cmd:**

****

**Solución en symbolab:**

** **