INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ALUMNO:

JONATHAN SAID GOMEZ MARBAN

PROFESOR:

JUAN JESÚS GUTIÉRREZ GARCÍA

FUNDAMENTOS DE PROGRAMACIÓN

PRÁCTICA 2 UNIDAD 2 (P22):

“Ecuaciones”

El propósito de esta práctica es resolver diez distintos problemas, todos ellos relacionados a las funciones o ecuaciones cuadráticas. En el documento subido a la plataforma, se explica que es lo que se requiere que haga, todo esto se debe hacer apoyándose de los archivos proporcionados por el profesor, los cuales son "ecu2.h" y "ecu2.c".

Problemas Propuestos

1. Dado una ecuación de segundo grado escribir sus raíces aclarando si son complejas o reales.
2. Sean c_1 y c_2 ecuaciones de segundo grado. Si c_1 y c_2 tiene sólo una raíz doble, r_1 y r_2 respectivamente, construir otra ecuación c_3 que tenga que tenga raíces r_1 y r_2.
3. Dados c_1 y c_2 ecuaciones de segundo grado calcular la suma de las dos y encontrar las raíces de la suma. En caso de que no existan mostrar un mensaje de error.
4. Se tiene una ecuación c_1 que sólo tiene dos raíces, modifica el valor constante (coeficiente c) para que sólo tenga una raíz.
5. Se tiene una ecuación c_1 que sólo tiene una raíz, modifica el valor constante (coeficiente c) para que tenga dos raíces reales. Muestra las nuevas raíces.
6. Se tiene una ecuación c_1 que tiene raíces complejas, modifica el coeficiente de la x para que tenga dos raíces reales. Muestra las nuevas raíces.
7. Verifica que si una ecuación de segundo grado tiene una raíz compleja su conjugado también es raíz de la ecuación. Para hacerlo construye una función que permita evaluar la ecuación en un valor complejo.
8. Dada una ecuación de segundo grado encuentra el valor extremo (máximo o mínimo)
9. Encuentra el punto medio de las raíces de cualquier ecuación de segundo grado. En caso de tener sólo una raíz el punto medio es la raíz. Si las raíces son complejas el punto medio lo calculas con las partes reales de las raíces.
10. Dado un valor real α y una ecuación de segundo grado $ax^2 + bx + c = 0$ con raíces reales, construye una nueva ecuación que sea $ax^2 + (2\alpha a + b)x + \alpha^2 a + b\alpha + c = 0$ y calcula las raíces de la nueva ecuación. ¿Qué relación hay entre las raíces originales y las nuevas? (en el programa verifica con $\alpha = -2, -1, 1, 2$).

Imagen 1. Lista de problemas.

Como se observa en la imagen 1, se detallan los ejercicios propuestos a resolver.

Se empieza explicando y presentando imágenes sobre cómo fue resuelto el primero de ellos. Se pide que se escriba la raíz de una ecuación aclarando si es compleja o real. Para ello se analizará la siguiente imagen perteneciente al archivo "problemas.c".

Cabe destacar que para cada función que se está definiendo en este archivo existe su prototipo en el archivo .h llamado de la misma forma.

```
void uno(double a, double b, double c) {
    ecu2 ecuacion;
    int resultado, ok;
    char salida[100];

    ok = crea_ecu2(&ecuacion, a, b, c);
    if (ok == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }

    printf("La ecuacion es %s\n", tostr_ecu2(salida, &ecuacion));
    resultado = raiztipo_ecu2(&ecuacion);
    if (resultado == RAIZ_COMPL) {
        printf("Tiene raices complejas\n");
    } else {
        printf("Tiene raices reales\n");
    }
    printf("\n");
}
```

Imagen 2. Función uno.

Se puede observar que en las primeras 2 líneas de código se declaran 3 variables. La primera de ellas es de tipo ecu2, la cual es un tipo que recibe 3 valores de tipo double, que representaran los coeficientes de a, b y c de nuestra ecuación, además de un cuarto campo el cual es el discriminante de la misma. Posteriormente de declaran 2 enteros, el primero de ellos nos ayudará a almacenar el tipo de raíz (doble, complejas o diferentes) que tiene la ecuación proporcionada a esta variable se le ha dado el nombre de "resultado"; por otro lado, la segunda variable de tipo entero nos permitirá almacenar, ya sea un código de error o bien un "OK", es decir, que con la función crea_ecu2, nosotros podremos asignarle los coeficientes a la ecuación y saber el valor del discriminante, dependiendo de este último, nos regresara un OK o un código de error, posteriormente esta variable "ok" de tipo entero, nos permitirá saber si la ecuación es cuadrática o no, a partir de compararla con una constante simbólica llamada COEFA_CERO, si la sentencia if resulta verdadero significa que el coeficiente de a es 0, y por tanto, la ecuación no es cuadrática y no tiene caso seguir con las siguientes líneas de código, por ello se ha colocado un return y el mensaje "no es una ecuación cuadrática". Todo lo anterior se repite en la gran mayoría de las funciones de este programa, por lo que, de ahora en adelante, simplemente lo expresaré de la siguiente forma: crear la ecuación, verificarla (que sea cuadrática) y obtener el tipo de raíz que tiene.

En la imagen 2 se logra observar que la función "uno", recibe 3 parámetros y no regresa nada, estos tres parámetros son los coeficientes que tendrá la ecuación. La variable salida

que es un arreglo de tipo carácter se usa como parámetro para la función tostr_ecu2 (una función perteneciente al archivo ecu2.c), la cual nos permite convertir la ecuación en una cadena.

Después que se almacena el tipo de raíz de la ecuación se procede a comparar la variable donde esta almacenado, "resultado", con una constante simbólica llamada RAIZ_COMPL, la cual nos permite saber si es una ecuación con raíz compleja, de ser así, la función mostrara el mensaje que aparece en la imagen 2 y de lo contrario el mensaje "tiene raíces reales".

Para el problema 2 se muestra la siguiente imagen:

```
void dos(double a1, double b1, double c1, double a2, double b2, double c2){
    ecu2 ecuacion1, ecuacion2;
    int ok1, ok2, resultado1, resultado2;

    ok1 = crea_ecu2(&ecuacion1, a1, b1, c1);
    if (ok1 == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    ok2 = crea_ecu2(&ecuacion2, a2, b2, c2);
    if (ok2 == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    resultado1 = raiztipo_ecu2(&ecuacion1);
    resultado2 = raiztipo_ecu2(&ecuacion2);
    if (resultado1 == RAIZ_DOBLE && resultado2 == RAIZ_DOBLE) {
        double raiz1,raiz2,raizNueva1,raizNueva2;
        double nueva_a=1;
        int ok3;
        raizpos_ecu2(&raiz1,&ecuacion1);
        raizpos_ecu2(&raiz2,&ecuacion2);
        ecu2 ecuacion3;
        ok3 = crea_ecu2(&ecuacion3, nueva_a, raiz1+raiz2, raiz1*raiz2);
        raizneg_ecu2(&raizNueva1,&ecuacion3);
        raizpos_ecu2(&raizNueva2,&ecuacion3);
        printf(" %f\n", raizNueva1);
        printf(" %f\n", raizNueva2);
    } else {
        printf(" Las raices introducidas no son raices dobles\n");
    }
}
```

Imagen 3. Función dos.

Como se logra observar la función dos recibe 6 parámetros, los cuales son los coeficientes de 2 ecuaciones cuadráticas distintas. En este programa se crean 2 ecuaciones, llamadas ecuación 1 y2 y lo mismo para las variables "ok" y "resultado", que tienen la misma finalidad de uso ya explicada con anterioridad. Por tanto, se crean 2 ecuaciones, ambas se verifican y se obtiene su tipo de raíz. Después se tiene la condición que si el "resultado1" y el "resultado2" (las cuales almacenan el tipo de raíz de ecuación 1 y 2 respectivamente) son iguales a la constante simbólica llamada "RAIZ_DOBLE", la cual, me indica que la ecuación cuadrática solo tiene una solución, si la sentencia es verdadera, se crearan 5 variables de tipo double y un entero, que se explicara su finalidad más adelante. Una vez declaradas las variables, con la función "raizpos_ecu2", la cual nos da la solución de ecuaciones que tienen raíces reales, le pasamos como parámetros por referencia a "raiz1", que es donde se guardara el valor de la solución de la ecuación, siendo así como segundo parámetro la "ecuacion1" y de la misma forma usamos "raizpos_ecu2" pasando la "raiz2" y "ecuacion2". Después creamos una nueva ecuación (de tipo ecu2), la cual es llamada "ecuacion3" y con la función "crea_ecu2" inicializaos los valores de nuestra "ecuacion3" y nos regresara un OK el cual lo almacenaremos en la variable "ok3"; Es pertinente explicar el razonamiento sobre como se crea la "ecuacion3" y como es que se resuelve el problema planteado que se muestra en la imagen 1. Lo que se busaca es que las soluciones de la ecuación 1 y 2 (las cuales solo tienen 1 solución, como se explicó), sean las soluciones para una nueva ecuación, la cual llamamos ecuación 3, para ello se muestra la siguiente imagen:

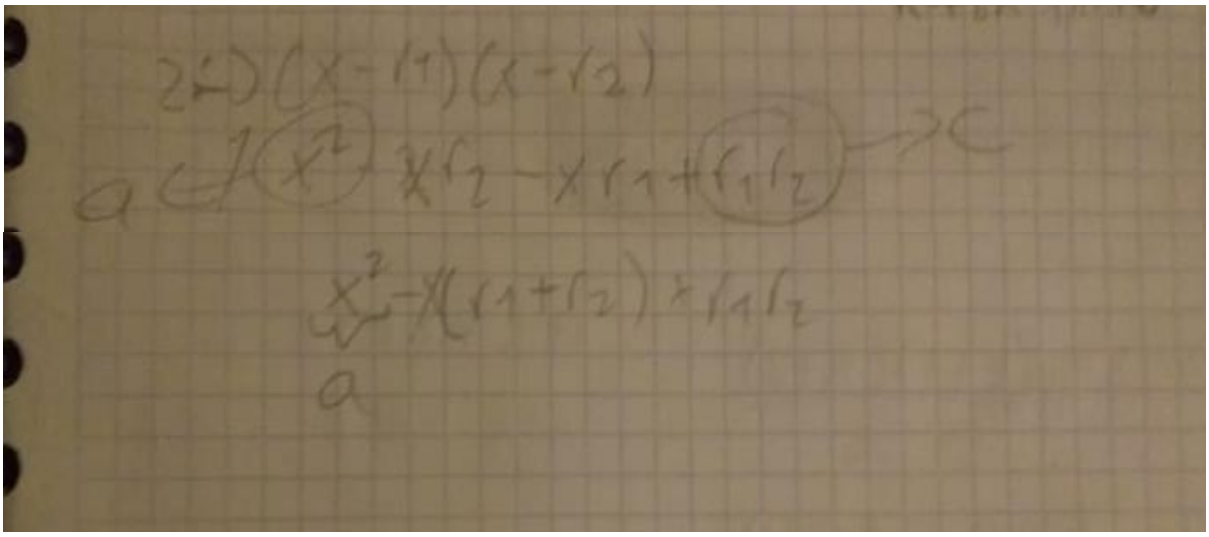

Imagen 4. Razonamiento para problema 2.

Se sabe que la soluciones de una ecuación cuadrática se pueden saber si se factoriza dicha ecuación cuadrática, tómese la factorización que aparece en la imagen 4 como las soluciones de nuestra ecuación 3, ya que queremos que raiz1 y raiz2 sean las soluciones de nuestra nueva ecuación, simplemente las colocamos dentro de cada binomio, como se

muestra en la imagen, posteriormente, se desarrolla para que nos quede de la forma ax^2+bx+c, y así poder determinar cuáles serán los valores de los coeficientes de la "ecuacion3", notamos que al desarrollar el valor para a tiene que ser 1, para b tiene que ser (raiz1+raiz2) y para c el producto de raiz1 y raiz2.

Es por ello por lo que al momento de inicializar los valores de la "ecuacion3", se tiene la variable "nueva_a" la cual vale 1, para el coeficiente de a; "raiz1+raiz2", para el coeficiente de b y finalmente para el coeficiente de c es "raiz1*raiz2".

Después con la función "raizpos_ecu2", la cual calcula la raíz positiva de la ecuación que le pasemos y la almacena en una variable, sacaremos la raíz positiva de "ecuacion3" y la almacenaremos en "raizNueva1", cabe mencionar que se hace paso de parámetros por referencia y de manera análoga, calcularemos la raíz negativa de la "ecuacion3" solo que con la función "raizneg_ecu2", y el valor de ella se guardara en "raizNueva2", posteriormente solo se imprimirán los valores de "raizNueva1" y "raizNueva2". Todo lo anterior está dentro de una sentencia if, es decir, si no se cumple la condición, pasará al else, la cual solo mostrará el mensaje que aparece en la imagen 3.

Problema 3

En este problema se busca resolver que, dadas dos ecuaciones cuadraditas, se sume cada campo de ellas, es decir, se suma el coeficiente a de la ecuación 1 con el coeficiente de a de la ecuación 2, así mismo con los coeficientes de b y c. Posteriormente se sacara la raíz de esa nueva ecuación propuesta, pero no sin antes analizar que tipo de raíz tiene, dependiendo de ello, podrá ser una solución compleja, una sola solución o bien dos soluciones distintas. Para ello se explicará la imagen 5 que se encuentra en la parte de abajo. La función tres recibe 6 parámetros, dentro de ella se crea una ecuación, la cual se llama "ecuacion3" de tipo ecu2, después se inicializan los coeficientes de la misma (note que aquí, ya se están sumando los coeficientes como se indicó), con la función "crea_ecu2", además de que me regresa un código de error o bien un OK, dependiendo del coeficiente de a, después se verifica que sea una ecuación cuadrática comparando su valor con la constante simbólica "COEFA_CERO", si la variable "ok" s igual a la constante, se imprimirá el mensaje "No es una ecuación cuadrática". Posteriormente obtendremos el tipo de raíz que tiene la "ecuacion3", después aparecen 3 sentencias if, en cada una de ellas se está comprando la variable "resultado", la cual tiene almacenada el tipo de raíz que tiene la ecuación, con alguna constante simbólica (que representan que tipo de raíz es la que se nos está presentando). Si "resultado" es igual a "RAIZ_COMPL", esto quiere decir que nuestro discriminante es negativo, por tanto las soluciones implican números imaginarios, para poder convertir las raíces de tipo compleja a reales se usa la función "raizreal_ecu2", la cual se hace paso de parámetros por referencia de la ecuación que tiene la raíz compleja y el otro parámetro será donde guardaremos el cálculo de la raíz parte real de la raíz compleja.

Posteriormente se imprimirá esa variable donde guardamos el valor de la raíz y se dará por finalizada la función, debido a que se coloca un return. De manera análoga si el "resultado" es igual a "RAIZ_DOBLE", entonces se ejecutará lo mencionado con anterioridad, pero en vez de usar la función "raizreal_ecu2" se usará "raizpos_ecu2" (la cual ya se ha explicado con anterioridad su funcionamiento) y se imprimirá el valor de la raíz. Ahora bien, si el "resultado" es igual a "RAIZ_DIFER", entonces se tendrá que utilizar las funciones "raizpos_ecu2" y "raizneg_ecu2", que dicho de manera simple, nos regresaran los valores de la raíz positiva y negativa respectivamente, haciendo paso de parámetros por referencia y nos regresara 2 raíces distintas, las cuales almacenamos en dos variables distintas que después imprimiremos.

```
58  void tres(double a1, double b1, double c1, double a2, double b2, double c2){
59      ecu2 ecuacion3;
60      int ok3,resultado;
61      ok3=crea_ecu2(&ecuacion3, a1+a2, b1+b2, c1+c2);
62      if (ok3 == COEFA_CERO) {
63          printf("No es una ecuacion cuadratica\n");
64          return;
65      }
66      resultado = raiztipo_ecu2(&ecuacion3);
67      if (resultado == RAIZ_COMPL) {
68          printf("Tiene raices complejas\n");
69          double raiz;
70          raizreal_ecu2(&raiz,&ecuacion3);
71          printf(" %f\n", raiz);
72          return ;
73      }
74      if (resultado == RAIZ_DIFER) {
75          printf("Tiene raices diferentes y reales\n");
76          double raiz1, raiz2;
77          raizneg_ecu2(&raiz1,&ecuacion3);
78          raizpos_ecu2(&raiz2,&ecuacion3);
79          printf(" %f\n", raiz1);
80          printf(" %f\n", raiz2);
81          return ;
82      }
83      if (resultado == RAIZ_DOBLE) {
84          printf("Tiene una sola raiz doble\n");
85          double raiz;
86          raizpos_ecu2(&raiz,&ecuacion3);
87          printf(" %f\n", raiz);
88      }
89  }
```

Imagen 5. Función tres.

Problema 4.

```c
void cuatro(double a, double b, double c){
    ecu2 ecuacion;
    int ok,resultado;
    ok=crea_ecu2(&ecuacion, a, b, c);
    if (ok == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    resultado=raiztipo_ecu2(&ecuacion);
    if (resultado == RAIZ_DOBLE) {
        printf("La ecuacion solo tiene una solucion\n");
        double raiz;
        raizpos_ecu2(&raiz,&ecuacion);
        printf(" %f\n", raiz);
    } else {
        printf("Las raices eran complejas o reales, pero distintas, se trabajo con ella, y ahora, solo tiene una raiz\n");
        double nueva_c = (b*b)/(4*a);
        double raiz2;
        ecu2 ecuacion2;
        ok=crea_ecu2(&ecuacion2, a, b, nueva_c);
        raizpos_ecu2(&raiz2,&ecuacion2);
        printf(" %f\n", raiz2);

    }
}
```


Imagen 6. Función cuatro.

En las primeras líneas de código de la imagen 6 se crea una ecuación (se inicializan los valores), se verifica que sea cuadrática y se obtiene su tipo de raíz, el cual es comparado con la constante simbólica llamada “RAÍZ_DOBLE”, si esto ocurre, procederemos a sacar el valor de la raíz a partir de la función “raizpos_ecu2” y después se imprimirá, si esto no ocurre, se declara una variable tipo double llamada “nueva_c”, la cual se inicializa como se nota en la imagen, con esta asignación a “nueva_c” garantizaremos que la nueva ecuación que crearemos tenga solo una raíz, esta asignación surge del razonamiento en que el discriminante tiene que ser 0 para que solo haya una solución, por tanto b^2=4ac, además, el problema 4 pide que solo modifiquemos el coeficiente c para que la ecuación que tenga dos raíces, solo tenga una, es por ello que se despeja c y queda de la forma en que se muestra en la imagen 6. Una vez hecho eso, se crea una nueva ecuación y se inicializan los valores, tomando el coeficiente de a, como el mismo que fue proporcionado, así como el b, el único que se cambia es el de c, por la variable “nueva_c”, posteriormente sacamos el valor de la raíz y por ultimo se imprime el valor de la raíz.

Problema 5

En el se busca que aquellas ecuaciones que tengan solo una raíz pasen a tener 2, únicamente modificando el valor del coeficiente c. Para ello, explicare el razonamiento que me ayudará a resolver este problema. Primero se nos dice que lo que buscamos es que la ecuación tenga 2 raíces, por lo que, si nos proporcionan una que ya las tenga, no habría que hacer otra cosa, mas que sacara el valor de ambas raíces e imprimirlas, luego, en caso de no

ser asi, entonces se creara un nuevo coeficiente para c, el cual valga 0, para que cuando se multiplique dentro del discriminante, se haga 0 y forzosamente me quede solamente con el valor de b^2 ,y así, la ecuación pueda tener más de dos soluciones, siempre y cuando el coeficiente de b sea distinto de 0. Se mostrará el código de la función cinco:

```
void cinco(double a, double b, double c){
    ecu2 ecuacion;
    int ok,resultado;
    ok=crea_ecu2(&ecuacion, a, b, c);
    if (ok == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    resultado = raiztipo_ecu2(&ecuacion);
    if (resultado == RAIZ_DIFER) {
        printf("Raiz diferente\n");
        double raiz1, raiz2;
        raizneg_ecu2(&raiz1,&ecuacion);
        raizpos_ecu2(&raiz2,&ecuacion);
        printf(" %f\n",raiz1);
        printf(" %f\n",raiz2);
    }else {
        printf("La raiz era compleja o era  una raiz igual, hemos hecho algunos cambios, y ahora, tiene 2 raices reales\n");
        double nueva_c=0;
        ecu2 ecuacion2;
        double raiz1,raiz2;
        ok=crea_ecu2(&ecuacion2, a, b, nueva_c);
        raizneg_ecu2(&raiz1,&ecuacion2);
        raizpos_ecu2(&raiz2,&ecuacion2);
        printf(" %f\n",raiz1);
        printf(" %f\n",raiz2);
    }
}
```

Imagen 7. Función cinco.

Se logra notar que se crea una ecuación, se verifica que sea cuadrática y posteriormente se obtiene su tipo de raíz, como se explicó, si el tipo de raíz es igual a “RAIZ_DIFER”, entonces solo habrá que usar las funciones las cuales me permiten sacar los valores de la raíz negativa y positiva, las cuales son “raizneg_ecu2” y “raizpos_ecu2” respectivamente, posteriormente solo se imprimen. En caso contrario se crea una variable llamada “nueva_c” que se inicializa en 0 (para lograr el razonamiento que se explicó anteriormente), además de que se crean “raiz1” y “raiz2”, que es donde guardaremos los valores de la raíz positiva y negativa, como se hizo anteriormente en la sentencia if, después, se crea una nueva ecuación llamada “ecuacion2” y se inicializan sus valores con la función “crea_ecu2”, alterando únicamente el coeficiente c, es decir, en vez de poner “c”, pondremos “nueva_c”, posteriormente sacamos los valores de las 2 raíces y los imprimimos finalmente.

Problema 6.

Se nos plantea el problema de que se recibe una ecuación con raíz de tipo compleja, una vez que se sabe si es compleja, entonces alterar el coeficiente b para que tenga raíces reales distintas. Entonces observemos el código de la función 6:

```c
void seis(double a, double b, double c){
    ecu2 ecuacion;
    int ok,resultado;
    ok=crea_ecu2(&ecuacion, a, b, c);
    if (ok == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    resultado = raiztipo_ecu2(&ecuacion);
    if (resultado == RAIZ_COMPL || resultado == RAIZ_DOBLE ) {
        printf("El polinomio tenia raiz compleja, se ha trabajado sobre ella, y ahora, tiene 2 raices reales\n");
        double raiz1,raiz2;
        double nueva_b=(4*a*c);
        ecu2 ecuacion2;
        ok=crea_ecu2(&ecuacion2, a, nueva_b, c);
        raizneg_ecu2(&raiz1,&ecuacion2);
        raizpos_ecu2(&raiz2,&ecuacion2);
        printf(" %f\n", raiz1);
        printf(" %f\n", raiz2);
        return ;
    }else {
        printf("El polinomio tiene 2 raices reales\n");
        double raiz1,raiz2;
        ok=crea_ecu2(&ecuacion, a, b, c);
        raizneg_ecu2(&raiz1,&ecuacion);
        raizpos_ecu2(&raiz2,&ecuacion);
        printf(" %f\n", raiz1);
        printf(" %f\n", raiz2);
        return ;
    }
}
```

Imagen 8. Función seis.

Se crea la ecuación, se inicializan los valores, se verifica que sea cuadrática y se obtiene su tipo de raíz. Una vez hecho esto, habrá dos casos distintos; primero: Si el "resultado" es igual a "RAIZ_COMPL" o "resultado" es igual a "RAIZ_DOBLE" y segundo: Si el "resultado" es igual a "RAIZ_DIFER". Analicemos el primer caso:
Se logra observar que se crean 4 variables. Se nota que "nueva_b" se inicializa, ahora bien, ¿Cómo se llega a esa asignación?, de la siguiente forma, ya sea que el tipo de raíz sea compleja o doble se necesita que la "nueva_b" sea mayor que 4ac, por tanto, si se igualan, significará que tienen al mismo valor, pero al momento de desarrollar el discriminante la "nueva_b" se eleva al cuadrado, dejándonos así, el discriminante mayor a 0 y por tanto, la ecuación nueva tendrá 2 raíces distintas. Entonces en ambos casos se crea una ecuación, se inicializan sus valores (poniendo "nueva_b" en vez de "b"), se sacan sus raíces, tanto positiva como negativa con "raizpos_ecu2" y "raizneg_ecu2". Si esto no ocurre entonces significa que la ecuación ya tiene dos raíces reales y por tanto solo hay que obtener e imprimir el valor de ellas.

Problema 8

Se pide encontrar el valor máximo o mínimo de una ecuación cuadrática, para ello se usa la formula y=(4ac-b^2)/(4a). Entonces, se mostrará como fue que se realizó.

```c
void ocho(double a, double b, double c){
    ecu2 ecuacion;
    int ok,resultado;
    ok=crea_ecu2(&ecuacion, a, b, c);
    if (ok == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    resultado = raiztipo_ecu2(&ecuacion);
    if (resultado == RAIZ_DOBLE) {
        printf("La raiz es doble, por lo tanto, el valor de la maxima o minima es 0\n");
        return ;
    }
    double valormin_max=(4*a*c-b*b)/(4*a);
    if(a>0) {
        printf("El valor minimo es: %f\n", valormin_max);
    }else {
        printf("El valor maximo es: %f\n", valormin_max);
    }
}
```

Imagen 9. Función ocho.

Se crea la ecuación, se inicializan los coeficientes, se verifica que sea cuadrática y se obtiene su tipo de raíz (todo esto explicado con anterioridad). Se sabe que, si el discriminante vale 0, entonces el valor máximo o mínimo de la ecuación cuadrática es 0, por tanto, se coloca la condición y si se cumple se termina la función; en caso contrario se inicializa la variable "valormin_max" y se le asigna la formula mencionada y se analizara una nueva sentencia if, en donde se determina si se esta calculando el valor máximo o el valor mínimo, todo dependerá del signo del coeficiente cuadrático (a), por tanto, si a es positiva, entonces se esta calculando el valor mínimo y si es negativa, entonces se esta calculando el valor máximo, en ambos casos se imprime la variable "valormin_max".

Problema 9.

Dada una ecuación y saber su tipo de raíz, se quiere calcular el punto medio, estableciendo lo siguiente:

1. Si la ecuación cuadrática tiene dos soluciones distintas, entonces el punto medio será el promedio de las raíces.
2. Si es una raíz doble o compleja, entonces el valor del punto medio será el valor de la raíz de la ecuación.

```c
void nueve(double a, double b, double c){
    ecu2 ecuacion;
    int ok,resultado;
    ok=crea_ecu2(&ecuacion, a, b, c);
    if (ok == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    resultado = raiztipo_ecu2(&ecuacion);
    if (resultado == RAIZ_DIFER) {
        double raiz1, raiz2, puntoMedio;
        raizneg_ecu2(&raiz1,&ecuacion);
        raizpos_ecu2(&raiz2,&ecuacion);
        puntoMedio=(raiz1+raiz2)/2;
        printf(" %f\n", puntoMedio);
    }
    if (resultado == RAIZ_DOBLE) {
        double raiz;
        raizpos_ecu2(&raiz,&ecuacion);
        printf(" %f\n", raiz);
    }
    if (resultado == RAIZ_COMPL) {
        double raiz;
        raizreal_ecu2(&raiz,&ecuacion);
        printf(" %f\n", raiz);
        return ;
    }
}
```

Imagen 10. Función nueve.

Se crea la ecuación, se inicializan los coeficientes, se verifica que sea cuadrática y se obtiene su tipo de raíz. La primer sentencia if compara el "resultado" con la constante simbólica "RAIZ_DIFER" (como se explicó en los untos establecidos), entonces se saca la raíz negativa y positiva con "raizneg_ecu2" y "raizpos_ecu2" respectivamente, después los valores de esas raíces se suman y se dividen entre 2 (el promedio), y finalmente el resultado se imprime.
En caso de que esto no suceda, puede significar que la raíz es doble o compleja, en ambos casos, se saca la raíz, para la raíz doble se ocupa "raizpos_ecu2" y para la compleja "raizreal_ecu2" y en ambos casos se imprimen.

Problema 10

Con una ecuación que tenga raíz o raíces reales y dado un valor real, que nombraremos alfa, construir una nueva ecuación que sea:

$ax^2 + (2\alpha a + b)x + \alpha\, 2a + b\alpha +$ c

Donde α es alfa. En la siguiente imagen se muestra el código.

```
void diez(double a, double b, double c, double alfa){
    ecu2 ecuacion;
    int ok,resultado;
    ok=crea_ecu2(&ecuacion, a, b, c);
    if (ok == COEFA_CERO) {
        printf("No es una ecuacion cuadratica\n");
        return;
    }
    resultado = raiztipo_ecu2(&ecuacion);
    if (resultado == RAIZ_DOBLE) {
        printf("Se ha introducido una ecuacion con raiz doble. ");
        double raizoriginal;
        raizpos_ecu2(&raizoriginal,&ecuacion);
        printf("Esta es la raiz original: %f\n", raizoriginal);
        double raiz1;
        ok=crea_ecu2(&ecuacion, a, (2*alfa*a)+b, (alfa*alfa*a)+(b*alfa)+c);
        raizneg_ecu2(&raiz1,&ecuacion);
        printf("Esta es la nueva raiz: %f\n", raiz1);
        printf("Este es el valor de alfa: %f y esta es la diferencia entre la raiz original y la nueva: %f\n", alfa, raiz1-raizoriginal);
        return ;
    }
    if (resultado == RAIZ_DIFER) {
        printf("Se ha introducido una ecuacion con raiz diferente. ");
        double raizoriginal1,raizoriginal2;
        raizpos_ecu2(&raizoriginal1,&ecuacion);
        raizneg_ecu2(&raizoriginal2,&ecuacion);
        printf("Estas son las raices originales: %f, %f\n", raizoriginal1, raizoriginal2);
        double raiz1,raiz2;
        ok=crea_ecu2(&ecuacion, a, (2*alfa*a)+b, (alfa*alfa*a)+(b*alfa)+c);
        raizpos_ecu2(&raiz1,&ecuacion);
        raizneg_ecu2(&raiz2,&ecuacion);
        printf("Esta son las nuevas raices: %f, %f\n", raiz1,raiz2);
        printf("Este es el valor de alfa: %f y esta es la diferencia entre la raiz original positiva y la nueva: %f\n", alfa, raiz1-raizoriginal1);
        printf("Este es el valor de alfa: %f y esta es la diferencia entre la raiz original negativa y la nueva: %f\n", alfa, raiz2-raizoriginal2);
        return ;
    }else {
        printf("Las raices introducidas son complejas");
    }
}
```

Imagen 11. Función 10.

Cabe aclarar que esta función tiene 4 parámetros, los cuales son: a, b, c y alfa.

Se crea la ecuación, se inicializan los coeficientes, se verifica que sea cuadrática y se obtiene su tipo de raíz. Su tipo de raíz se compara con la constante simbólica "RAÍZ_DOBLE", se declara una variable llamada "raizoriginal", luego se saca el valor de esa raíz (con "raizpos_ecu2") y se imprime. Posteriormente se declara "raiz1", se usa "crea_ecu2" para inicializar los valores de la ecuación (tal y como se explicó anteriormente) y se saca la raíz negativa (con "raizneg_ecu2"). Posteriormente se imprime el valor de la raíz negativa, junto con la especificación del valor de alfa y se nota que la diferencia entre la raíz original y la nueva es el valor con signo cambiado de alfa.

Ocurre de manera muy similar si el tipo de raíz es diferente (2 soluciones), solo que se crean 2 raíces originales y 2 nuevas raíces (la positiva y la negativa). En caso de que ninguno de los dos casos ocurra, mandara el mensaje que se muestra en la imagen 11.

Con esto, se ha explicado todo el archivo problemas.c; Presentaré la imagen del archivo.h donde solo vienen contenidos los prototipos de todas las funciones ya explicadas.

```c
#ifndef PROBLEMAS_H
#define PROBLEMAS_H

void uno(double a, double b, double c);
void dos(double a1, double b1, double c1, double a2, double b2, double c2);
void tres(double a1, double b1, double c1, double a2, double b2, double c2);
void cuatro(double a, double b, double c);
void cinco(double a, double b, double c);
void seis(double a, double b, double c);
void ocho(double a, double b, double c);
void nueve(double a, double b, double c);
void diez(double a, double b, double c, double alfa);



#endif
```

Imagen 12. Problemas.h.

Finalmente mostraré como está compuesto el usa_problemas.c y luego como todas estas funciones se compilan, y que resuelven el problema propuesto.

```c
#include "ecu2.h"
#include "problemas.h"
#include <stdio.h>

int main() {
    printf("Problema 1: \n");
    uno(-4, -2, 2); // reales
    uno(36, -216, 648); // complejas
    uno(0, 1, 1); // no cuadraticas
    printf("\n")

    printf("Problema 2: \n");
    dos(1,-6,9,1,-10,25);  //dobles
    dos(1,4,5,1,8,1);  //2,12,6
    dos(1,4,2,2,2,1);  //3,6,3
    dos(0,1,2,0,2,1);  //0,3,3
    printf("\n")

    printf("Problema 3: \n");
    tres(2,4,5,1,8,9);  //3,12,14, compleja
    tres(1,4,5,1,8,1);  //2,12,6
    tres(1,4,2,2,2,1);  //3,6,3
    tres(0,1,2,0,2,1);  //0,3,3
    printf("\n")

    printf("Problema 4: \n");
    cuatro(2,4,5);
    cuatro(3,6,3);
    cuatro(2,12,6);
    cuatro(0,1,2);
    printf("\n")

    printf("Problema 5: \n");
    cinco(2,4,5);
    cinco(3,6,3);
    cinco(2,12,6);
    cinco(0,1,2);
    printf("\n")
```

Imagen 13. Invocando de la función uno a cinco.

Cabe mencionar que, en la mayoría de las invocaciones, se colocan valores para que generen raíces de tipo doble, diferente o real, o bien que no sea una función cuadrática.

```
    printf("Problema 6: \n");
    seis(2,4,5);
    seis(3,6,3);
    seis(2,12,6);
    seis(0,1,2);
    printf("\n");

    printf("Problema 8: \n");
    ocho(1,-3,2);
    ocho(3,6,3);
    ocho(-2,12,6);
    ocho(0,1,2);
    printf("\n");

    printf("Problema 9: \n");
    nueve(2,4,5);
    nueve(3,6,3);
    nueve(2,12,6);
    nueve(0,1,2);
    printf("\n");

    printf("Problema 10: \n");
    diez(2,4,2,1);
    diez(3,6,3,2);
    diez(1,-6,9,-1);
    diez(1,-10,25,-2);
    diez(0,25,-2,-2);
    printf("\n");

}
```

Imagen 14. Invocación de la función seis a diez.

Ahora se procede a compilarlo sin que genere warnings, y cumpliendo con el propósito:

```
C:\Users\jona-\Downloads\Ec_2>gcc problemas.c usa_problemas.c ecu2.c

C:\Users\jona-\Downloads\Ec_2>a
Problema 1:
La ecuacion es -4.00x^2-2.00x+2.00=0
Tiene raices reales

La ecuacion es 36.00x^2-216.00x+648.00=0
Tiene raices complejas

No es una ecuacion cuadratica

Problema 2:
 -5.000000
 -3.000000
 Las raices introducidas no son raices dobles
 Las raices introducidas no son raices dobles
No es una ecuacion cuadratica

Problema 3:
Tiene raices complejas
 -2.000000
Tiene raices diferentes y reales
 -5.449490
 -0.550510
Tiene una sola raiz doble
 -1.000000
No es una ecuacion cuadratica

Problema 4:
Las raices eran complejas o reales, pero distintas, se trabajo con ella, y ahora, solo tiene una raiz
 -1.000000
La ecuacion solo tiene una solucion
 -1.000000
Las raices eran complejas o reales, pero distintas, se trabajo con ella, y ahora, solo tiene una raiz
 -3.000000
No es una ecuacion cuadratica

Problema 5:
La raiz era compleja o era  una raiz igual, hemos hecho algunos cambios, y ahora, tiene 2 raices reales
 -2.000000
 0.000000
La raiz era compleja o era  una raiz igual, hemos hecho algunos cambios, y ahora, tiene 2 raices reales
 -2.000000
 0.000000
Raiz diferente
 -5.449490
 -0.550510
No es una ecuacion cuadratica
```

Imagen 15. Compilar y ejecutar (uno a cinco).

Como se logra observar en la imagen 15, usando el comando gcc y agregando los archivos usa_problemas.c, problemas.c y ecu2.c compilaremos.

El resto de las pruebas aparecerán en la siguiente imagen:

```
Problema 6:
El polinomio tenia raiz compleja, se ha trabajado sobre ella, y ahora, tiene 2 raices reales
 -19.874209
 -0.125791
El polinomio tenia raiz compleja, se ha trabajado sobre ella, y ahora, tiene 2 raices reales
 -11.916080
 -0.083920
El polinomio tiene 2 raices reales
 -5.449490
 -0.550510
No es una ecuacion cuadratica

Problema 8:
El valor minimo es: -0.250000
La raiz es doble, por lo tanto, el valor de la maxima o minima es 0
El valor maximo es: 24.000000
No es una ecuacion cuadratica

Problema 9:
 -1.000000
 -1.000000
 -3.000000
No es una ecuacion cuadratica

Problema 10:
Se ha introducido una ecuacion con raiz doble. Esta es la raiz original: -1.000000
Esta es la nueva raiz: -2.000000
Este es el valor de alfa: 1.000000 y esta es la diferencia entre la raiz original y la nueva: -1.000000
Se ha introducido una ecuacion con raiz doble. Esta es la raiz original: -1.000000
Esta es la nueva raiz: -3.000000
Este es el valor de alfa: 2.000000 y esta es la diferencia entre la raiz original y la nueva: -2.000000
Se ha introducido una ecuacion con raiz doble. Esta es la raiz original: 3.000000
Esta es la nueva raiz: 4.000000
Este es el valor de alfa: -1.000000 y esta es la diferencia entre la raiz original y la nueva: 1.000000
Se ha introducido una ecuacion con raiz doble. Esta es la raiz original: 5.000000
Esta es la nueva raiz: 7.000000
Este es el valor de alfa: -2.000000 y esta es la diferencia entre la raiz original y la nueva: 2.000000
No es una ecuacion cuadratica


C:\Users\jona-\Downloads\Ec_2>
```

Imagen 16. Compilar y ejecutar (seis a diez).

Con esto se da por concluido la practica y el reporte de la misma.