**Universidad Católica Andrés Bello**

**Facultad de Ingeniería informática**

**Cátedra de Algoritmos y Programación I**

**Docentes: Franklin Bello y Zulma Díaz**

Alumnos:

Edwin Rodríguez – C.I: 30437298

Juan Quijada – C.I: 30212474

Práctica #2 – Estructuras de selección.

Problema N°1.

Una empresa extranjera de aviación fumiga cosechas contra una gran variedad de plagas. Los valores cobrados a los granjeros dependen de lo que éste desea fumigar y de cuántas hectáreas se fumigan, de acuerdo a la siguiente distribución:

Tipo 1: fumigación contra malas hierbas, $10 por hectárea.

Tipo 2: fumigación contra langostas, $15 hectárea.

Tipo 3: fumigación contra malas gusanos, $20 por hectárea.

Tipo 4: fumigación contra todo lo anterior, $30 por hectárea.

Si el área a fumigar es mayor de 1000 hectáreas, el granjero goza de un 5% de descuento. Además, cualquier granjero cuya cuenta sobrepase los $3000 se le descuenta un 10% sobre la cantidad que exceda dicho precio.

Si hay que aplicar ambos conceptos, el correspondiente a la superficie se considera primero.

Por cada pedido se tiene la siguiente información: nombre del granjero, tipo de fumigación solicitada (1,2,3,4) y el número de hectáreas a fumigar.

Para cada solicitud el programa debe mostrar el nombre del granjero y valor total a pagar.

**Análisis del problema**

|  |  |  |
| --- | --- | --- |
| Entrada | Proceso | Salida |
| El nombre del granjero (Nombre)  El tipo de fumigación que desea el granjero (tipo)  La cantidad de Hectáreas que se van a fumigar (Area) | Pedir los datos de entrada.  Calcular el precio del servicio a partir de las hectáreas y el tipo deseado (Precio)  Verificar si aplican los descuentos correspondientes  (si área>1000; si precio>3000)  Mostrar los datos de salida | Mostrar el nombre del granjero, el monto  (Nombre, Precio) |

**Algoritmo en Pseudocódigo**

Programa: EmpresaDeFumigacion;

Constantes

  Tipo1 = 10;

  Tipo2 = 15;

  Tipo3 = 20;

  Tipo4 = 30;

  DscArea = 0.05;

  DscPrecio = 0.1;

Variables

  Tipo: Caracter;

  Precio, Area: Real;

  Cost\_Hect: Entero;

  Nombre: String;

**Inicio**

  Escribir('Introduzca el nombre del que solicita el servicio');

  Leer(Nombre);

  Escribir('Introduzca el tipo de fumigación que se quiere realizar');

  Escribir('1: Fumigación contra malas hierbas');

  Escribir('2: Fumigación contra langostas');

  Escribir('3: Fumigación contra gusanos');

  Escribir('4: Fumigación contra todo lo anterior');

  Leer(Tipo);

  Escribir(‘Introduzca la cantidad de hectáreas que quiere tratar con la fumigación');

  Leer(area);

  Caso de (Tipo) haga

    1: Cost\_Hect := Tipo1;

    2: Cost\_Hect := Tipo2;

    3: Cost\_Hect := Tipo3;

    4: Cost\_Hect := Tipo4;

  Fin-caso

  Precio = Area \* Cost\_Hect;

  Si (Area > 1000) entonces

    Precio := Precio - (Precio \* DscArea);

Fin-si

  Si (Precio > 3000) entonces

    Precio := Precio - ((Precio-3000) \* DscPrecio);

Fin-si

  Escribir('Servicio para el granjero: ', Nombre);

  Escribir('Total a pagar: $', Precio);

**Fin**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Código Fuente**

Program EmpresaDeFumigacion;

Uses Crt;

Const

  //Se crean constantes para cada tipo de fumigacion

  Tipo1 = 10;

  Tipo2 = 15;

  Tipo3 = 20;

  Tipo4 = 30;

  DscArea = 0.05;  //Descuento por Area; 5%

  DscPrecio = 0.1; //Descuento por Precio; 10%

Var

  Tipo: Char;

  Precio, Area: Real;

  Cost\_Hect: Integer; //Costo por Hectarea.

  Nombre: String;

Begin

  writeln('Introduzca el nombre del que solicita el servicio');

  readln(Nombre);

  writeln('Introduzca el tipo de fumigacion que se quiere realizar');

  writeln('1: Fumigacion contra malas hierbas');

  writeln('2: Fumigacion contra langostas');

  writeln('3: Fumigacion contra gusanos');

  writeln('4: Fumigacion contra todo lo anterior');

  readln(Tipo);

  writeln(‘Introduzca la cantidad de hectareas que quiere tratar con la fumigacion');

  readln(area);

  Case Tipo Of

    '1': Cost\_Hect := Tipo1;

    '2': Cost\_Hect := Tipo2;

    '3': Cost\_Hect := Tipo3;

    '4': Cost\_Hect := Tipo4;

  End;

  Precio := Area \* Cost\_Hect;

  //Se verifican los casos para aplicar cada descuento, colocados segun prioridad

  If Area > 1000 Then

    Precio := Precio - (Precio \* DscArea);

  If Precio > 3000 Then

    Precio := Precio - ((Precio-3000) \* DscPrecio);

  //Se imprime el resultado

Writeln();

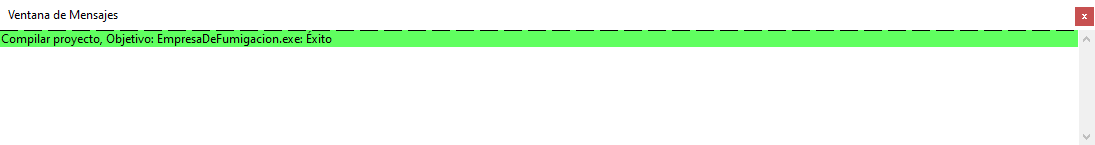
  Writeln('Servicio para el granjero ', Nombre);

  writeln('Total a pagar: $', Precio:0:2);

  readkey();

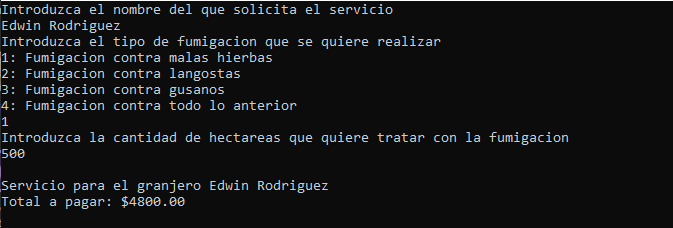
End.

**Resultado de la compilación:**

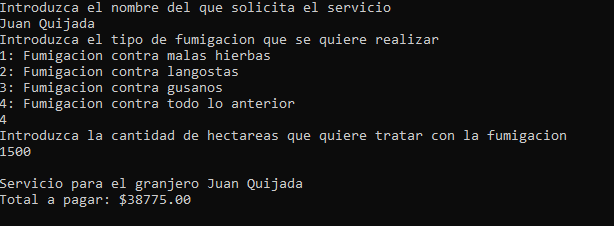


**Ejecución del programa:**

**Caso 1**



Caso 2



Problema N°2

Elaborar un programa en Pascal que realice el cálculo de las RAICES en una ecuación cuadrática, según lo descrito a continuación.

Para calcular las raíces de la ecuación cuadrática se usa la fórmula: ax2 + bx + c = 0 , teniendo en cuenta que :

a) Si “a” es igual a 0 y “b” es igual a 0, imprimiremos un mensaje diciendo que la “Ecuación es degenerada”.

b) Si “a” es igual a 0 y “b” no es igual a 0, existe una raíz única con valor –c/b, en este caso se debe enviar el mensaje “Existe una raíz única” y de seguida el valor de la raíz.

c) En los demás casos, utilizaremos la fórmula: x = (-b ± √(b2 – 4ac))/(2a)

* La expresión d = b2 – 4ac se denomina discriminante.
* Si “d” es mayor o igual que 0 entonces hay dos raíces reales, que se deben calcular según la fórmula y mostrarse, junto al mensaje “Existen dos raíces reales”.
* Si “d” es menor que 0 entonces hay dos raíces complejas de la forma: J+Qi y J-Qi. Siendo J = –b/2a y Q = √ABS(b2-4ac)/(2a), que se lee como ( la raíz cuadrada del valor absoluto de (b2-4ac))/(2a).

Nota : el carácter √ corresponde a raíz cuadrada

**Análisis del problema**

|  |  |  |
| --- | --- | --- |
| Entrada | Proceso | Salida |
| Los términos de la ecuación cuadrática de la forma ax^2+bx+c=0 (a, b, c) | Pedir datos de entrada.  Verificar qué condiciones cumplen los términos ingresados (a, b, c)  Obtener la raíz o raíces de la ecuación cuadrática dependiendo de las condiciones cumplidas (Raiz1, Raiz2)  Mostrar el o los resultados obtenidos. | Mostrar la raíz o raíces de la ecuación cuadrática (raiz1,Raiz2 o solo Raiz1), si “la ecuación es degenerada”, o si tiene dos raíces complejas (parte real “L” y parte imaginaria “Q”) |

**Algoritmo en Pseudocódigo**

Programa: EcuacionCuadratica

Variables

a,b,c,d,J,Q,Raiz1,Raiz2: real;

**Inicio**

  Escribir('Este programa calcula las raices de una ecuacion cuadrática.');

  Escribir('Introduzca los terminos "a", "b" y "c", en ese orden especifico');

  Leer(a,b,c);

  Si (a=0) y (b=0) entonces

    Escribir('La ecuacion es degenerada.');

  Sino

      Si (a=0) y (b<>0) entonces

          Raiz1 := -c/b;

          Escribir ('Existe una raiz unica: ', Raiz1);

      Sino

          d = b\*b - 4\*a\*c;

          Si (d>0) entonces

              Raiz1 := (-b+ Raíz cuadrada de (d))/(2\*a);

              Raiz2 := (-b- Raíz cuadrada de (d))/(2\*a);

              Escribir('Existen dos raíces reales: ', Raiz1,' y ', Raiz2);

          Sino

              Si (d<0) entonces

                  J := -b/(2\*a);

                  Q := Raiz cuadrada de (Valor absoluto de (b\*b - 4\*a\*c))/(2\*a);

Si Q<0 entonces

Q:=Q\*-1;

Fin-si;

                  Escribir('Existen dos raíces complejas: (', J,')+(', Q, 'i) y (', J, ')-(', Q,'i)');

              Sino

                  Raiz1 := -b/(2\*a);

                  Escribir('Existe una raiz unica: ',Raiz1);

              Fin-si;

          Fin-si;

      Fin-si;

  Fin-si;

**Fin**

**Código fuente**

Program EcuacionCuadratica;

Uses crt;

Var

  a,b,c,d,J,Q,Raiz1,Raiz2: real;

//a,b,c: Terminos de la ecuacion cuadratica; d: Discriminante; j,Q: Parte real e imaginaria de las raices complejas

Begin

  //Se piden los datos de entrada al usuario

Writeln('Este programa calcula las raices de una ecuacion cuadratica de la forma "ax^2 + bx + c = 0');

Writeln('Para poderlas calcular, introduzca los terminos "a", "b" y "c", en ese orden especifico');

  Readln(a,b,c);

// Se crea un condicional para cada posibilidad de las variables de la ecuacion cuadratica.

  If (a=0) And (b=0) Then

    Writeln('La ecuacion es degenerada')

  Else

    Begin

      If (a=0) And (b<>0) Then

        Begin

          Raiz1 := -c/b;

          Writeln('Existe una raiz unica: ',Raiz1:0:2);

        End

      Else

        Begin

          d := b\*b - 4\*a\*c;

          If (d>0) Then

            Begin

              Raiz1 := (-b+sqrt(d))/(2\*a);

              Raiz2 := (-b-sqrt(d))/(2\*a);

              Writeln('Existen dos raices reales: ',Raiz1:0:2,' y ',Raiz2:0:2);

            End

          Else

            Begin

              If (d<0) Then

                Begin

                  //Aquí se utiliza la "i" para marcar la parte imaginaria del resultado

                  J := -b/(2\*a);

                  Q := sqrt(abs(b\*b - 4\*a\*c))/(2\*a);

//Se multiplica Q por -1 si es negativo, para volverlo positivo y evitar la multiplicacion de signos

If Q<0 then

Q:=Q\*-1;

  Writeln('Existen dos raices complejas: (',J:0:2,')+(',Q:0:2,'i) y (',J:0:2,')-(',Q:0:2,'i)');

                End

              Else

                Begin

                  Raiz1 := -b/(2\*a);

                  Writeln('Existe una raiz unica: ',Raiz1:0:2);

                End;

            End;

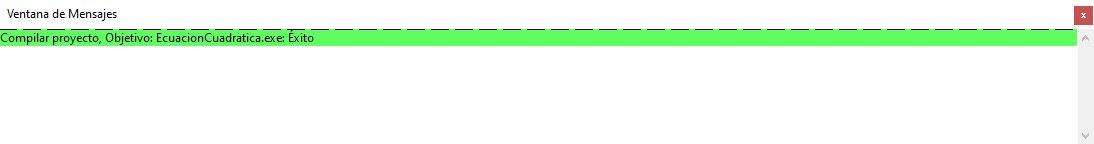
        End;

    End;

  Readkey();

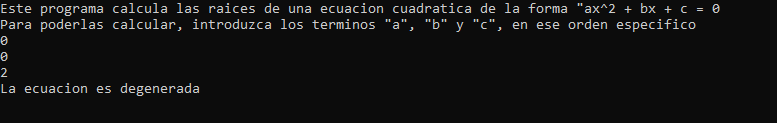
End.

**Resultado de la compilación:**

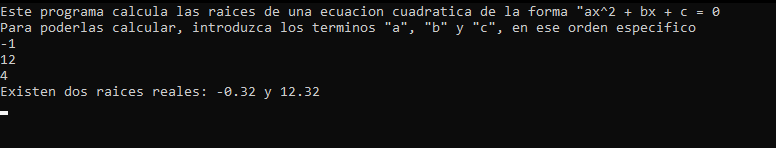


**Ejecución del programa:**

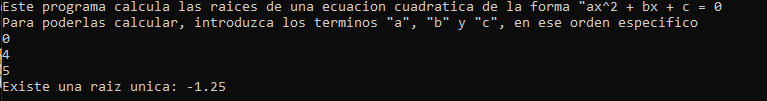
Caso 1:



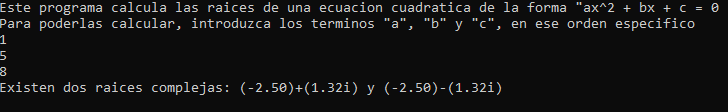
Caso 2:

****

Caso 3:



Caso 4:



Caso 5:

