

## 12 NijiHouteishiki.dpr (整係数二次方程式の解)

係数が整数の二次方程式  $ax^2 + bx + c = 0$  の解を求めます。

### 12.1 実行例

整係数二次方程式  $ax^2 + bx + c = 0$  の解を求めます

係数 a b c [a = 0 を入れると終わります] ? 1 1 1	$-1/2 \pm 3/2 i$
係数 a b c [a = 0 を入れると終わります] ? 1 2 3	$-1 \pm 2 i$
係数 a b c [a = 0 を入れると終わります] ? 1 0 3	$\pm 3 i$
係数 a b c [a = 0 を入れると終わります] ? 3 0 1	$\pm 3/3 i$
係数 a b c [a = 0 を入れると終わります] ? 4 4 17	$-1/2 \pm 2 i$
係数 a b c [a = 0 を入れると終わります] ? -3 2 -1	$1/3 \pm 2/3 i$
係数 a b c [a = 0 を入れると終わります] ? 1 4 4	-2
係数 a b c [a = 0 を入れると終わります] ? 9 -6 1	1/3
係数 a b c [a = 0 を入れると終わります] ? 1 2 -3	1, -3
係数 a b c [a = 0 を入れると終わります] ? 3 2 -1	1/3, -1
係数 a b c [a = 0 を入れると終わります] ? 2 3 -1	$-3/4 \pm 17/4$
係数 a b c [a = 0 を入れると終わります] ? 2 4 -1	$-1 \pm 6/2$
係数 a b c [a = 0 を入れると終わります] ? 2 0 -1	$\pm 2/2$
係数 a b c [a = 0 を入れると終わります] ? 1 4 2	$-2 \pm 2$
係数 a b c [a = 0 を入れると終わります] ? -3 8 4	$4/3 \pm 2 7/3$
係数 a b c [a = 0 を入れると終わります] ? 0 0 0	

## 12.2 未完成プログラム

メインルーチンと、係数を読み込み  $a$  が 0 でないか調べる関数と、虚数解の場合の解を書くプロシージャができています。その他の足りない部分を補って完成させなさい。メインルーチンは、係数を読んで、判別式 (discriminant)  $D = b^2 - 4ac$  で場合分けして、下請けに出します。

```

program NijiHouteishiki; // 学生証番号 氏名
{$APPTYPE CONSOLE}
uses SysUtils;

procedure Root(N : Integer; var P,Q : Integer);
    { n = 整数 p q }
    { ただし, n > 0, q は 1 より大きい平方数で割り切れない }
begin
end; {Root}

function Gcd(M,N : Integer) : Integer;
    { m と n の最大公約数 (Greatest Common Divisor) }
begin
end; {Gcd}

function Lcm(M,N : Integer) : Integer;
    { m と n の最小公倍数 (Least Common Multiple) }
begin
end; {Lcm}

procedure Yakubun(Si1,Bo1 : Integer; var Si2,Bo2 : Integer);
    { 約分 (reduce) }
    { Si1/Bo1 = 約分 Si2/Bo2 }
    { Bo2 > 0 になるようにする }
begin
end; {Yakubun}

procedure AddBunsuu(Si1,Bo1,Si2,Bo2 : Integer; var Si3,Bo3 : Integer);
    { 足し算 (add) }
    { Si1/Bo1 + Si2/Bo2 Si3/Bo3 }
begin
end; {AddBunsuu}

procedure WriteBunsuu(Si,Bo : Integer);
    { 分数を書く }
    { 整数は分子だけ書く }
begin
end; {WriteBunsuu}

procedure Kyosuukai(A,B,D : Integer);
    { 虚数解の場合 }
var
    Buns1,Bunbo1 : Integer;
    Buns2,Bunbo2 : Integer;
    P,Q : Integer;
begin
    Root(-D,P,Q);
    Yakubun(-B,2*A,Buns1,Bunbo1);
    Yakubun(P,Abs(2*A),Buns2,Bunbo2);
    if Buns1 <> 0
    then WriteBunsuu(Buns1,Bunbo1);
    Write(' ± ');

```

```

if Q = 1
then begin
    WriteBunsuu(Bunsi2,Bunbo2);
end
else begin
    if Bunsi2 > 1
    then Write(Bunsi2);
    Write(' ', Q);
    if Bunbo2 > 1
    then Write('/', Bunbo2);
end;
WriteLn(' i ');
end; {KyoSuukai}

procedure Juukai(A,B : Integer);
    { 重解の場合 }
begin
end; {Juukai}

procedure Yuurisuukai(A,B,D : Integer);
    { 有理数解の場合 }
begin
end; {Yuurisuukai}

procedure Murisuukai(A,B,D : Integer);
    { 無理数解の場合 }
begin
end; {Murisuukai}

function ReadKeisuu(var A,B,C : Integer) : Boolean;
    { 係数 a , b , c を読む }
    { 戻り値 = True ( a ≠ 0 のとき ) }
    {           False ( a = 0 のとき ) }
begin
    WriteLn;
    Write(' 係数 a b c [a = 0 を入れると終わります] ? ');
    ReadLn(A,B,C);
    ReadKeisuu := A<>0;
end; {ReadKeisuu}

var
    A,B,C,D : Integer;
begin {Main}
    WriteLn(' 整数係数二次方程式  $ax^2 + bx + c = 0$  の解を求めます');
    while ReadKeisuu(A,B,C) do
    begin
        Write('':55);
        D := Sqr(B)-4*A*C;
        if D < 0
        then KyoSuukai(A,B,D)
        else if D = 0
            then Juukai(A,B)
            else if D = Sqr(Trunc(Sqrt(D)))
                then Yuurisuukai(A,B,D)
                else Murisuukai(A,B,D);
    end;
end.

```