

```

int start()
{
//偶数か否かのチェック
if(MathMod(Calculate_Bars,2) == 0 && Even_Number == false) ← ア
{
Alert("Parameter must be Odd Number.");
Even_Number = true; ← イ
return(0);
}

//計算範囲の選択
if(IndicatorCounted() == 0)
{
Calculated = 0;
}
else if(IndicatorCounted() > 0) ← (2)
{
Calculated = 1;
}

switch(Calculated) ← (3)
{
//1ティック目の計算
case 0:

for(int i = Bars; i >= 0; i--) ← ア
{
Support_Point[i] = NULL; ← イ
Resistance_Point[i] = NULL; ← イ

//サポートラインとレジスタンスラインの位置
int Center_Index = MathFloor(Calculate_Bars / 2) + i;
int Lowest_Index = iLowest(NULL,0,MODE_LOW,Calculate_Bars,i); ← ウ
int Highest_Index = iHighest(NULL,0,MODE_HIGH,Calculate_Bars,i);

if(Center_Index == Lowest_Index)
{
Support_Point[Center_Index] = Low[Center_Index];
}
if(Center_Index == Highest_Index) ← エ
{
Resistance_Point[Center_Index] = High[Center_Index];
}
}

i = Bars; ← オ
while(i >= 0) ← カ
{
//サポートライン
if(Support_Point[i] != NULL)
{
Support[i] = Support_Point[i];
}
else if(Support_Point[i] == NULL && Support[i+1] > 0) ← キ
{
Support[i] = Support[i+1];
}

//レジスタンスライン
if(Resistance_Point[i] != NULL)
{
Resistance[i] = Resistance_Point[i];
}
else if(Resistance_Point[i] == NULL && Resistance[i+1] > 0) ← ク
{
Resistance[i] = Resistance[i+1];
}

i--; ← ケ
}

break;

//2ティック目以降の計算
case 1:

Support_Point[0] = NULL; ← ア
Resistance_Point[0] = NULL; ← ア

//サポートラインとレジスタンスラインの位置
Center_Index = MathFloor(Calculate_Bars / 2);
Lowest_Index = iLowest(NULL,0,MODE_LOW,Calculate_Bars,0); ← イ
Highest_Index = iHighest(NULL,0,MODE_HIGH,Calculate_Bars,0);

if(Center_Index == Lowest_Index)
{
Support_Point[Center_Index] = Low[Center_Index];
}
if(Center_Index == Highest_Index) ← エ
{
Resistance_Point[Center_Index] = High[Center_Index];
}

for(i = Calculate_Bars - 1; i >= 0; i--) ← オ
{
//サポートライン
if(Support_Point[i] != NULL)
{
Support[i] = Support_Point[i];
}
else if(Support_Point[i] == NULL && Support[i+1] > 0) ← カ
{
Support[i] = Support[i+1];
}

//レジスタンスライン
if(Resistance_Point[i] != NULL)
{
Resistance[i] = Resistance_Point[i];
}
else if(Resistance_Point[i] == NULL && Resistance[i+1] > 0) ← キ
{
Resistance[i] = Resistance[i+1];
}
}

break;
}

return(0);
}

```

(1)

(2)

(3)

(4)

(5)