```
//Support-Resistance.mq4

#property indicator_chart_window

#property indicator_buffers 2                  #property命令を記述

#property indicator_color1 Aqua
#property indicator_color2 Magenta

//インジケータバッファーの宣言
double Support[];
double Resistance[];                           インジケーターバッファーを宣言
double Support_Point[];
double Resistance_Point[];

//変数の宣言
extern int Calculate_Bars = 5;
int Calculated            = 0;                 変数を宣言
bool Even_Number          = false;

int init()
{
  IndicatorBuffers(4);

  //インジケーターの値の精度
  IndicatorDigits(MarketInfo(Symbol(),MODE_DIGITS));

  //インジケーターバッファーのインデックス
  SetIndexBuffer(0,Support);
  SetIndexBuffer(1,Resistance);
  SetIndexBuffer(2,Support_Point);
  SetIndexBuffer(3,Resistance_Point);

  //インジケーターのラベル                        基本設定を記述
  SetIndexLabel(0,"Support");
  SetIndexLabel(1,"Resistance");

  //インジケーターのスタイル
  SetIndexStyle(0,DRAW_ARROW);
  SetIndexArrow(0,159);
  SetIndexStyle(1,DRAW_ARROW);
  SetIndexArrow(1,159);
  SetIndexStyle(2,DRAW_NONE);
  SetIndexStyle(3,DRAW_NONE);

  return(0);
}

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

      Even_Number = true;

      return(0);
    }

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

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

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

                 //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);
                 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)
                   {
                     Resistance_Point[Center_Index] = High[Center_Index];
                   }
               }

           i = Bars;

           while(i >= 0)
                 {
                   //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)
                           {
                             Support[i] = Support[i+1];
                           }

                   //レジスタンスライン                                    具体的な処理内容
                   if(Resistance_Point[i] != NULL)                        を記述
                     {
                       Resistance[i] = Resistance_Point[i];
                     }
                   else if(Resistance_Point[i] == NULL && Resistance[i+1] > 0)
                           {
                             Resistance[i] = Resistance[i+1];
                           }

                   i--;
                 }

           break;

           //２ティック目以降の計算
           case 1:

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

           //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);
           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)
             {
               Resistance_Point[Center_Index] = High[Center_Index];
             }

           for(i = Calculate_Bars - 1; i >= 0; i--)
               {
                 //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)
                         {
                           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)
                         {
                           Resistance[i] = Resistance[i+1];
                         }
               }

           break;
          }

  return(0);
}
```