```
#property copyright "Copyright 2025, Æea©"
#property link      "https:t.me/faderBoard"
#property version   "1.00"
#property strict
int OnInit()
  {
  OnReInit();
   return(INIT_SUCCEEDED);
  }
void OnDeinit(const int reason)
  {
  }
input int Commssion=0;
double com=Commssion*Point;
input int StopLoss=0;
double SL=StopLoss*Point;
input int TakeProfit=0;
double TP=TakeProfit*Point;
input double lot=0.01;
input int slip=100;
input int max=60;
input int min=3;
int x=max+2;
input bool Cc = true;
input bool cC = true;
input bool invert = true;
bool KC = invert;
int y=min-2;
int j;
double signal = 0;
double spread = Ask - Bid;
double cA[];
double cADX;
double mSO;
double sSO;
double iSO;
double aRVI;
double bRVI;
double cRVI;
double cAC;
double cForce;
double cOBV;
double cAD;
double cMFI;
double cMomentum;
double cDM;
double cWPR;
double cCCI;
double cRSI;
double iA[];
double iATR;
double iStdDev;
double iADX;
double mStochastic;
double sStochastic;
double iStochastic;
double mRVI;
double sRVI;
double iRVI;
double iAC;
double iForce;
double iOBV;
double iAD;
double iMFI;
double iMomentum;
double iDM;
double iWPR;
double iCCI;
double iRSI;
double iIHKt;
double iIHKk;
double kA[];
double lA[];
double IHKk[];
double IHKt[];
double RSI[];
double CCI[];
double MOM[];
double AD[];
double OBV[];
double Force[];
double MFI[];
double DeM[];
double RVIm[];
double AC[];
double StdDev[];
double ATR[];
double ADX[];
double Suply;
double iSuply;
double Demand;
double iDemand;
double f=100*(2.0/3);
double g=100*(1.0/3);
double gf=100*((2.0/5)/3);
int m;
int n;
string Regime[];
static double Premium[];
static double Discount[];
static double HH[];
static double LL[];
bool k[];
bool l[];
bool R=true;
bool U[];
double bSL;
double sSL;
double bTP;
double sTP;
int lOrder_id=-1;
int kOrder_id=-1;
int Buy=-1;
int Sell=-1;
bool A=true;
bool B=true;
bool a=true;
bool b=true;
bool ab=false;
static double D;
static double E;
static double p;
static double q;
bool K=false;
bool c=cC;
bool C=Cc;
bool u=false;
bool v=false;
bool iC=Cc;
bool jC=Cc;
static int Z=y+1;
static int z=y+1;
static int O=y+1;
static int o=y+1;
static int r;
static int W=y+1;
static int w=y+1;
static int I;
static int iI;
static int J;
static int iJ;
static int ij;
static int h;
static int toll=0;
string tally="";
bool tickTock=false;
//Open[2]
double iopen;
static int iZ=y+1;
static int iz=y+1;
static int iW=y+1;
static int iw=y+1;
static int iO=y+1;
static int io=y+1;
static int ir;
int S=x;
int T=x;
int X=y;
int Y=y;
bool FG=false;
bool GF=false;
double price;
double Price;
double open;
double iH;
double iL;
double Sale;
double iSale;
double Stock;
double iStock;
static datetime t;
void OnReInit()
    {
    KC = invert;
    ArrayInitialize(cA, 0); //ArrayResize(cA, 0);
    ArrayInitialize(iA, 0); //ArrayResize(iA, 0);
    ArrayInitialize(kA, 0); //ArrayResize(kA, 0);
    ArrayInitialize(lA, 0); //ArrayResize(lA, 0);
    ArrayInitialize(IHKk, 0); //ArrayResize(IHKk, 0);
    ArrayInitialize(IHKt, 0); //ArrayResize(IHKt, 0);
    ArrayInitialize(RSI, 0); //ArrayResize(RSI, 0);
    ArrayInitialize(CCI, 0); //ArrayResize(CCI, 0);
    ArrayInitialize(MOM, 0); //ArrayResize(MOM, 0);
    ArrayInitialize(AD, 0); //ArrayResize(AD, 0);
    ArrayInitialize(OBV, 0); //ArrayResize(OBV, 0);
    ArrayInitialize(Force, 0); //ArrayResize(Force, 0);
    ArrayInitialize(MFI, 0); //ArrayResize(MFI, 0);
    ArrayInitialize(DeM, 0); //ArrayResize(DeM, 0);
    ArrayInitialize(RVIm, 0); //ArrayResize(RVIm, 0);
    ArrayInitialize(AC, 0); //ArrayResize(AC, 0);
    ArrayInitialize(StdDev, 0); //ArrayResize(StdDev, 0);
    ArrayInitialize(ATR, 0); //ArrayResize(ATR, 0);
    ArrayInitialize(ADX, 0); //ArrayResize(ADX, 0);
    ArrayInitialize(StdDev, 0); //ArrayResize(StdDev, 0);
    /*ArrayInitialize(Regime, "");*/ArrayResize(Regime, 0);
    ArrayInitialize(Premium, 0); //ArrayResize(Premium, 0);
    ArrayInitialize(Discount, 0); //ArrayResize(Discount, 0);
    ArrayInitialize(HH, 0); //ArrayResize(HH, 0);
    ArrayInitialize(LL, 0); //ArrayResize(LL, 0);
    ArrayInitialize(k, 0); //ArrayResize(k, 0);
    ArrayInitialize(l, 0); //ArrayResize(l, 0);
    ArrayInitialize(U, 0); //ArrayResize(U, 0);
    R=true;
    ab=false;
    D=0;
    E=0;
    K=false;
    Z=y+1;
    z=y+1;
    O=y+1;
    o=y+1;
    r=0;
    W=y+1;
    w=y+1;
    I=0;
    iI=0;
    J=0;
    iJ=0;
    ij=0;
    h=0;
    toll=0;
    tally="";
    tickTock=false;
    iZ=y+1;
    iz=y+1;
    iW=y+1;
    iw=y+1;
    iO=y+1;
    io=y+1;
    ir=0;
    S=x;
    T=x;
    X=y;
    Y=y;
    FG=false;
    Print("ReSettled");
    }
void Unify()
    {
    ArrayResize(ATR,j+1);
    for(int i=0;i<j+1; i++){ATR[i]=iATR(NULL,0,j,i);}
    double maxATR=ATR[ArrayMaximum(ATR,WHOLE_ARRAY,0)];
    double minATR=ATR[ArrayMinimum(ATR,WHOLE_ARRAY,0)];
    double rangeATR=maxATR-minATR;
    if(rangeATR!=0) iATR=100*((iATR(NULL,0,j,0)-minATR)/rangeATR);
    ArrayResize(StdDev,j+1);
    for(int i=0;i<j+1; i++)
{StdDev[i]=iStdDev(NULL,0,j,0,MODE_SMA,PRICE_CLOSE,i);}
    double maxSD=StdDev[ArrayMaximum(StdDev,WHOLE_ARRAY,0)];
    double minSD=StdDev[ArrayMinimum(StdDev,WHOLE_ARRAY,0)];
    double rangeSD=maxSD-minSD;
    if(rangeSD!=0) iStdDev=100*((iStdDev(NULL,0,j,0,MODE_SMA,PRICE_CLOSE,0)-
minSD)/rangeSD);
    }
void Normalize()
    {
    Suply=iBands(NULL,0,j,2,0,PRICE_CLOSE,MODE_UPPER,0);
    iSuply=iBands(NULL,0,j,2,0,PRICE_CLOSE,MODE_UPPER,1);
    Demand=iBands(NULL,0,j,2,0,PRICE_CLOSE,MODE_LOWER,0);
    iDemand=iBands(NULL,0,j,2,0,PRICE_CLOSE,MODE_LOWER,1);
    ArrayResize(iA,13*((S+1)-Y));
    ArrayResize(cA,13*((S+1)-Y));
    double uADX[];
    ArrayResize(uADX,j+1);
    for(int i=0;i<j+1; i++){uADX[i]=iADX(NULL,0,j,PRICE_CLOSE,MODE_PLUSDI,i);}
    double maxuADX=uADX[ArrayMaximum(uADX,WHOLE_ARRAY,0)];
    double minuADX=uADX[ArrayMinimum(uADX,WHOLE_ARRAY,0)];
    double lADX[];
    ArrayResize(lADX,j+1);
    for(int i=0;i<j+1; i++)
{lADX[i]=iADX(NULL,0,j,PRICE_CLOSE,MODE_MINUSDI,i);}
    double maxlADX=lADX[ArrayMaximum(lADX,WHOLE_ARRAY,0)];
    double minlADX=lADX[ArrayMinimum(lADX,WHOLE_ARRAY,0)];
    ArrayResize(ADX,j+1);
    for(int i=0;i<j+1; i++){ADX[i]=iADX(NULL,0,j,PRICE_CLOSE,MODE_MAIN,i);}
    double maxmADX=ADX[ArrayMaximum(ADX,WHOLE_ARRAY,0)];
    double minmADX=ADX[ArrayMinimum(ADX,WHOLE_ARRAY,0)];
    double maxADX=MathMax(maxmADX,MathMax(maxuADX,maxlADX));
    double minADX=MathMin(minmADX,MathMin(minuADX,minlADX));
    double rangeADX=maxADX-minADX;
    if(rangeADX!=0)
        {
        iADX=MathAbs(100*((iADX(NULL,0,j,PRICE_CLOSE,MODE_MAIN,0)-
minADX)/rangeADX));
        iA[0*(S-Y)+(j-(Y+1))]=iADX;
        cADX=MathAbs(100*((ADX[1]-minADX)/rangeADX));
        cA[0*(S-Y)+(j-(Y+1))]=cADX;
        }
    int jSO=(int)MathRound((double)j*3.0/5);
    mStochastic=iStochastic(NULL,0,j,3,jSO,MODE_SMA,0,MODE_MAIN,0);
    sStochastic=iStochastic(NULL,0,j,3,jSO,MODE_SMA,0,MODE_SIGNAL,0);
    iStochastic=(mStochastic+sStochastic)/2;
    iA[1*(S-Y)+(j-(Y+1))]=iStochastic;
    mSO=iStochastic(NULL,0,j,3,jSO,MODE_SMA,0,MODE_MAIN,1);
    sSO=iStochastic(NULL,0,j,3,jSO,MODE_SMA,0,MODE_SIGNAL,1);
    iSO=(mSO+sSO)/2;
    cA[1*(S-Y)+(j-(Y+1))]=iSO;
    ArrayResize(RVIm,j+1);
    for(int i=0;i<j+1; i++){RVIm[i]=iRVI(NULL,0,j,MODE_MAIN,i);}
    double maxMRVI=RVIm[ArrayMaximum(RVIm,WHOLE_ARRAY,0)];
    double minMRVI=RVIm[ArrayMinimum(RVIm,WHOLE_ARRAY,0)];
    double RVIs[];
    ArrayResize(RVIs,j+1);
    for(int i=0;i<j+1; i++){RVIs[i]=iRVI(NULL,0,j,MODE_SIGNAL,i);}
    double maxSRVI=RVIs[ArrayMaximum(RVIs,WHOLE_ARRAY,0)];
    double minSRVI=RVIs[ArrayMinimum(RVIs,WHOLE_ARRAY,0)];
    double maxRVI=MathMax(maxMRVI,maxSRVI);
    double minRVI=MathMin(minMRVI,minSRVI);
    double rangeRVI=maxRVI-minRVI;
    if(rangeRVI!=0)
        {
        mRVI=100*((iRVI(NULL,0,j,MODE_MAIN,0)-minRVI)/rangeRVI);
        sRVI=100*((iRVI(NULL,0,j,MODE_SIGNAL,0)-minRVI)/rangeRVI);
        iRVI=(mRVI+sRVI)/2;
        iA[2*(S-Y)+(j-(Y+1))]=iRVI;
        aRVI=100*((iRVI(NULL,0,j,MODE_MAIN,1)-minRVI)/rangeRVI);
        bRVI=100*((iRVI(NULL,0,j,MODE_SIGNAL,1)-minRVI)/rangeRVI);
        cRVI=(aRVI+bRVI)/2;
        cA[2*(S-Y)+(j-(Y+1))]=cRVI;
        }
    ArrayResize(AC,j+1);
    for(int i=0;i<j+1; i++){AC[i]=iAC(NULL,0,i);}
    double maxAC=AC[ArrayMaximum(AC,WHOLE_ARRAY,0)];
    double minAC=AC[ArrayMinimum(AC,WHOLE_ARRAY,0)];
    double rangeAC=maxAC-minAC;
    if(rangeAC!=0)
        {
        iAC=MathAbs(100*((iAC(NULL,0,0)-minAC)/rangeAC));
        iA[3*(S-Y)+(j-(Y+1))]=iAC;
        cAC=MathAbs(100*((iAC(NULL,0,1)-minAC)/rangeAC));
        cA[3*(S-Y)+(j-(Y+1))]=cAC;
        }
    ArrayResize(Force,j+1);
    for(int i=0;i<j+1; i++){Force[i]=iForce(NULL,0,j,MODE_SMA,PRICE_CLOSE,i);}
    double maxForce=Force[ArrayMaximum(Force,WHOLE_ARRAY,0)];
    double minForce=Force[ArrayMinimum(Force,WHOLE_ARRAY,0)];
    double rangeForce=maxForce-minForce;
    if(rangeForce!=0)
        {
        iForce=100*((iForce(NULL,0,j,MODE_SMA,PRICE_CLOSE,0)-
minForce)/rangeForce);
        iA[4*(S-Y)+(j-(Y+1))]=iForce;
        cForce=100*((iForce(NULL,0,j,MODE_SMA,PRICE_CLOSE,1)-
minForce)/rangeForce);
        cA[4*(S-Y)+(j-(Y+1))]=cForce;
        }
    ArrayResize(OBV,j+1);
    for(int i=0;i<j+1; i++){OBV[i]=iOBV(NULL,0,PRICE_CLOSE,i);}
    double maxOBV=OBV[ArrayMaximum(OBV,WHOLE_ARRAY,0)];
    double minOBV=OBV[ArrayMinimum(OBV,WHOLE_ARRAY,0)];
    double rangeOBV=maxOBV-minOBV;
    if(rangeOBV!=0)
        {
        iOBV=100*((iOBV(NULL,0,PRICE_CLOSE,0)-minOBV)/rangeOBV);
        iA[5*(S-Y)+(j-(Y+1))]=iOBV;
        cOBV=100*((iOBV(NULL,0,PRICE_CLOSE,1)-minOBV)/rangeOBV);
        cA[5*(S-Y)+(j-(Y+1))]=cOBV;
        }
    ArrayResize(AD,j+1);
    for(int i=0;i<j+1; i++){AD[i]=iAD(NULL,0,i);}
    double maxAD=AD[ArrayMaximum(AD,WHOLE_ARRAY,0)];
    double minAD=AD[ArrayMinimum(AD,WHOLE_ARRAY,0)];
    double rangeAD=maxAD-minAD;
    if(rangeAD!=0)
        {
        iAD=100*((iAD(NULL,0,0)-minAD)/rangeAD);
```

```
          iA[6*(S-Y)+(j-(Y+1))]=iAD;
          cAD=100*((iAD(NULL,0,1)-minAD)/rangeAD);
          cA[6*(S-Y)+(j-(Y+1))]=cAD;
          }
      ArrayResize(MFI,j+1);
      for(int i=0;i<j+1; i++){MFI[i]=iMFI(NULL,0,j,i);}
      double maxMFI=MFI[ArrayMaximum(MFI,WHOLE_ARRAY,0)];
      double minMFI=MFI[ArrayMinimum(MFI,WHOLE_ARRAY,0)];
      double rangeMFI=maxMFI-minMFI;
      if(rangeMFI!=0)
          {
          iMFI=100*((iMFI(NULL,0,j,0)-minMFI)/rangeMFI);
          iA[7*(S-Y)+(j-(Y+1))]=iMFI;
          cMFI=100*((iMFI(NULL,0,j,1)-minMFI)/rangeMFI);
          cA[7*(S-Y)+(j-(Y+1))]=cMFI;
          }
      ArrayResize(MOM,j+1);
      for(int i=0;i<j+1; i++){MOM[i]=iMomentum(NULL,0,j,PRICE_CLOSE,i);}
      double maxMOM=MOM[ArrayMaximum(MOM,WHOLE_ARRAY,0)];
      double minMOM=MOM[ArrayMinimum(MOM,WHOLE_ARRAY,0)];
      double rangeMOM=maxMOM-minMOM;
      if(rangeMOM!=0)
          {
          iMomentum=100*((iMomentum(NULL,0,j,PRICE_CLOSE,0)-minMOM)/rangeMOM);
          iA[8*(S-Y)+(j-(Y+1))]=iMomentum;
          cMomentum=100*((iMomentum(NULL,0,j,PRICE_CLOSE,1)-minMOM)/rangeMOM);
          cA[8*(S-Y)+(j-(Y+1))]=cMomentum;
          }
      ArrayResize(DeM,j+1);
      for(int i=0;i<j+1; i++){DeM[i]=iDeMarker(NULL,0,j,i);}
      double maxDM=DeM[ArrayMaximum(DeM,WHOLE_ARRAY,0)];
      double minDM=DeM[ArrayMinimum(DeM,WHOLE_ARRAY,0)];
      double rangeDM=maxDM-minDM;
      if(rangeDM!=0)
          {
          iDM=100*(iDeMarker(NULL,0,j,0)-minDM)/rangeDM;
          iA[9*(S-Y)+(j-(Y+1))]=iDM;
          cDM=100*(iDeMarker(NULL,0,j,1)-minDM)/rangeDM;
          cA[9*(S-Y)+(j-(Y+1))]=cDM;
          }
      iWPR=iWPR(NULL,0,j,0)+100;
      iA[10*(S-Y)+(j-(Y+1))]=iWPR;
      cWPR=iWPR(NULL,0,j,1)+100;
      cA[10*(S-Y)+(j-(Y+1))]=cWPR;
      ArrayResize(CCI,j+1);
      for(int i=0;i<j+1; i++){CCI[i]=iCCI(Symbol(),0,j,PRICE_TYPICAL,i);}
```

```
    double maxCCI=CCI[ArrayMaximum(CCI,WHOLE_ARRAY,0)];
    double minCCI=CCI[ArrayMinimum(CCI,WHOLE_ARRAY,0)];
    double rangeCCI=maxCCI-minCCI;
    if(rangeCCI!=0)
        {
        iCCI=100*((iCCI(Symbol(),0,j,PRICE_TYPICAL,0)-minCCI)/rangeCCI);
        iA[11*(S-Y)+(j-(Y+1))]=iCCI;
        cCCI=100*((iCCI(Symbol(),0,j,PRICE_TYPICAL,1)-minCCI)/rangeCCI);
        cA[11*(S-Y)+(j-(Y+1))]=cCCI;
        }
    ArrayResize(RSI,j+1);
    for(int i=0;i<j+1; i++){RSI[i]=iRSI(NULL,0,j,PRICE_CLOSE,i);}
    double maxRSI=RSI[ArrayMaximum(RSI,WHOLE_ARRAY,0)];
    double minRSI=RSI[ArrayMinimum(RSI,WHOLE_ARRAY,0)];
    double rangeRSI=maxRSI-minRSI;
    if(rangeRSI!=0)
        {
        iRSI=100*((iRSI(NULL,0,j,PRICE_CLOSE,0)-minRSI)/rangeRSI);
        iA[12*(S-Y)+(j-(Y+1))]=iRSI;
        cRSI=100*((iRSI(NULL,0,j,PRICE_CLOSE,1)-minRSI)/rangeRSI);
        cA[12*(S-Y)+(j-(Y+1))]=cRSI;
        }
    int kIHK=(int)MathRound((double)j/2);
    int tIHK=(int)MathRound(((double)kIHK+1)/3);
    double IHKa[];
    double IHKb[];
    double IHKc[];
    ArrayResize(IHKa,j+1);
    for(int i=0;i<j+1; i++)
{IHKa[i]=iIchimoku(NULL,0,tIHK,kIHK,j,MODE_SENKOUSPANA,i);}
    double maxIHKa=IHKa[ArrayMaximum(IHKa,WHOLE_ARRAY,0)];
    double minIHKa=IHKa[ArrayMinimum(IHKa,WHOLE_ARRAY,0)];
    ArrayResize(IHKb,j+1);
    for(int i=0;i<j+1; i++)
{IHKb[i]=iIchimoku(NULL,0,tIHK,kIHK,j,MODE_SENKOUSPANB,i); }
    double maxIHKb=IHKb[ArrayMaximum(IHKb,WHOLE_ARRAY,0)];
    double minIHKb=IHKb[ArrayMinimum(IHKb,WHOLE_ARRAY,0)];
    ArrayResize(IHKc,j+1);
    for(int i=0;i<j+1; i++)
{IHKc[i]=iIchimoku(NULL,0,tIHK,kIHK,j,MODE_CHIKOUSPAN,26+i);}
    double maxIHKc=IHKc[ArrayMaximum(IHKc,WHOLE_ARRAY,0)];
    double minIHKc=IHKc[ArrayMinimum(IHKc,WHOLE_ARRAY,0)];
    ArrayResize(IHKt,j+1);
    for(int i=0;i<j+1; i++)
{IHKt[i]=iIchimoku(NULL,0,tIHK,kIHK,j,MODE_TENKANSEN,i);}
    double maxIHKt=IHKt[ArrayMaximum(IHKt,WHOLE_ARRAY,0)];
    double minIHKt=IHKt[ArrayMinimum(IHKt,WHOLE_ARRAY,0)];
    ArrayResize(IHKk,j+1);
    for(int i=0;i<j+1; i++)
{IHKk[i]=iIchimoku(NULL,0,tIHK,kIHK,j,MODE_KIJUNSEN,i);}
    double maxIHKk=IHKk[ArrayMaximum(IHKk,WHOLE_ARRAY,0)];
    double minIHKk=IHKk[ArrayMinimum(IHKk,WHOLE_ARRAY,0)];
    double
maxIHK=MathMax(maxIHKa,MathMax(maxIHKb,MathMax(maxIHKc,MathMax(maxIHKk,maxIHKt
))));
    double
minIHK=MathMin(minIHKa,MathMin(minIHKb,MathMin(minIHKc,MathMin(minIHKk,minIHKt
))));
    double rangeIHK=maxIHK-minIHK;
    if(rangeIHK!=0)
        {
        iIHKk=100*((iIchimoku(NULL,0,tIHK,kIHK,j,MODE_KIJUNSEN,0)-
minIHK)/rangeIHK);
        iIHKt=100*((iIchimoku(NULL,0,tIHK,kIHK,j,MODE_TENKANSEN,0)-
minIHK)/rangeIHK);
        }
    }
void M()
    {
    for(int i=0;i<13; i++)
        {
        if(Price>HH[j-(y+1)]) if((iA[i*(S-Y)+(j-(Y+1))]>f+gf)||(cA[i*(S-Y)+(j-
(Y+1))]<kA[i*(S-Y)+(j-(Y+1))])) m++;
        else if(price>HH[j-(y+1)]) if((iA[i*(S-Y)+(j-(Y+1))]>f+gf)||(iA[i*(S-
Y)+(j-(Y+1))]<kA[i*(S-Y)+(j-(Y+1))])) m++;
        else if(iA[i*(S-Y)+(j-(Y+1))]>f+gf) m++;
        }
    if((iA[0*(S-Y)+(j-(Y+1))]>f+gf)||(iA[0*(S-Y)+(j-(Y+1))]<g-gf)) m++;
    if((iIHKt>f+gf)&&(iIHKk>f+gf)) m++;
    if(Price>HH[j-(y+1)])
        {
        ArrayResize(kA,13*(S-Y));
        for(int i=0;i<13; i++){kA[i*(S-Y)+(j-(Y+1))]=cA[i*(S-Y)+(j-(Y+1))];}
        HH[j-(y+1)]=Price;
        }
    }
void N()
    {
    for(int i=0;i<13; i++)
        {
        if(Price<LL[j-(y+1)]) if((iA[i*(S-Y)+(j-(Y+1))]<g-gf)||(cA[i*(S-Y)+(j-
(Y+1))]>lA[i*(S-Y)+(j-(Y+1))])) n++;
```

```
        else if(price<LL[j-(y+1)]) if((iA[i*(S-Y)+(j-(Y+1))]<g-gf)||(iA[i*(S-
Y)+(j-(Y+1))]>lA[i*(S-Y)+(j-(Y+1))])) n++;
        else if(iA[i*(S-Y)+(j-(Y+1))]<g-gf) n++;
        }
    if((iA[0*(S-Y)+(j-(Y+1))]>f+gf)||(iA[0*(S-Y)+(j-(Y+1))]<g-gf)) n++;
    if((iIHKt<g-gf)&&(iIHKk<g-gf)) n++;
    if(Price<LL[j-(y+1)])
        {
        ArrayResize(lA,13*(S-Y));
        for(int i=0;i<13; i++){lA[i*(S-Y)+(j-(Y+1))]=cA[i*(S-Y)+(j-(Y+1))];}
        LL[j-(y+1)]=Price;
        }
    }
void F()
    {
    Normalize();
    if(j==h) ab=false;
    k[j-(y+1)]=false;
    l[j-(y+1)]=false;
    HH[j-(y+1)]=iH;
    LL[j-(y+1)]=iL;
    Premium[j-(y+1)]=iH;
    Discount[j-(y+1)]=iL;
    ArrayResize(kA,13*(S-Y));
    ArrayResize(lA,13*(S-Y));
    for(int i=0;i<13; i++)
        {
        kA[i*(S-Y)+(j-(Y+1))]=cA[i*(S-Y)+(j-(Y+1))];
        lA[i*(S-Y)+(j-(Y+1))]=cA[i*(S-Y)+(j-(Y+1))];
        }
    if((R==true)&&(FG==true))
        {
        ArrayResize(U,x-y);
        int V=0; U[j-(y+1)]=true;
        for(int i=y+1;i<x; i++){if(U[i-(y+1)]==true) V++;}
        if(V==x-y){R=false;} V=0;
        }
    }
void G()
    {
    double H=iHigh(Symbol(), Period(), 1);
    double L=iLow(Symbol(), Period(), 1);
    ArrayResize(kA,13*(S-Y));
    ArrayResize(lA,13*(S-Y));
    for(j=2;j<h+1; j++)
        {
```

```
        if(j==x) break;
        k[j-(y+1)]=false;
        l[j-(y+1)]=false;
        HH[j-(y+1)]=H;
        LL[j-(y+1)]=L;
        Premium[j-(y+1)]=H;
        Discount[j-(y+1)]=L;
        for(int i=0;i<13; i++)
            {
            kA[i*(S-Y)+(j-(Y+1))]=cA[i*(S-Y)+(j-(Y+1))];
            lA[i*(S-Y)+(j-(Y+1))]=cA[i*(S-Y)+(j-(Y+1))];
            }
        }
    }
void S()
    {
    if(SL!=0)
        {
        sSL=Bid+SL-com;
        bSL=Ask-SL+com;
        }
    if(TP!=0)
        {
        sTP=Bid-TP;
        bTP=Ask+TP;
        }
    }
void T()
    {
    if(((b==false)&&(lOrder_id!=-1))||((a==false)&&(kOrder_id!=-1)))
        {
        Buy=lOrder_id; Sell=kOrder_id;
        }
    else if(((b==false)&&(kOrder_id!=-1))||((a==false)&&(lOrder_id!=-1)))
        {
        Buy=kOrder_id; Sell=lOrder_id;
        }
    if(Buy!=-1)
        {
        if(OrderSelect(Buy,SELECT_BY_TICKET))
            {
            /*E=OrderOpenPrice();*/ E=price; q=E+3*com;
            }
        }
    else if(Sell!=-1)
        {
```

```
        if(OrderSelect(Sell,SELECT_BY_TICKET))
            {
            /*D=OrderOpenPrice();*/ D=price; p=D-3*com;
            }
        }
    if((K==false)&&((SL!=0)||(com!=0)))
        {
        if((b==false)&&(Price>q))
            {
            b=OrderModify(Buy,E,E+com,bTP,0,CLR_NONE); K=true;
            }
        else if((a==false)&&(Price<p))
            {
            a=OrderModify(Sell,D,D-com,sTP,0,CLR_NONE); K=true;
            }
        }
    if((E!=0)&&(price>=E/*+com*/)) B=true;
    else if((E!=0)&&(price<E/*+com*/)) B=false;
    if((D!=0)&&(price<=D/*-com*/)) A=true;
    else if((D!=0)&&(price>D/*-com*/)) A=false;
    }
void A()
    {
    if((v==true)&&(lOrder_id!=-1))
        {
        int bTrade=OrderClose(lOrder_id,lot,Bid,slip,Blue);
        lOrder_id=-1;
        }
    else if((v==true)&&(kOrder_id!=-1))
        {
        int bTrade=OrderClose(kOrder_id,lot,Bid,slip,Blue);
        kOrder_id=-1;
        }
    E=0; B=false; K=false; Buy=-1;
    }
void B()
    {
    if((u==true)&&(kOrder_id!=-1))
        {
        int sTrade=OrderClose(kOrder_id,lot,Ask,slip,Red);
        kOrder_id=-1;
        }
    else if((u==true)&&(lOrder_id!=-1))
        {
        int sTrade=OrderClose(lOrder_id,lot,Ask,slip,Red);
        lOrder_id=-1;
        }
    D=0; A=false; K=false; Sell=-1;
    }
void P()
    {
    S();
    if(C==true)
        {

lOrder_id=OrderSend(_Symbol,OP_BUY,lot,Ask,slip,bSL,bTP,"EA",1992470,0,Blue);
        b=false;
        u=false;
        v=true;
        }
    else
        {

lOrder_id=OrderSend(_Symbol,OP_SELL,lot,Bid,slip,sSL,sTP,"EA",1992470,0,Red);
        a=false;
        u=true;
        v=false;
        }
    }
void Q()
    {
    S();
    if(C==true)
        {

kOrder_id=OrderSend(_Symbol,OP_SELL,lot,Bid,slip,sSL,sTP,"EA",1992470,0,Red);
        a=false;
        u=true;
        v=false;
        }
    else
        {

kOrder_id=OrderSend(_Symbol,OP_BUY,lot,Ask,slip,bSL,bTP,"EA",1992470,0,Blue);
        b=false;
        u=false;
        v=true;
        }
    }
void H(){M(); if(m>=12) k[j-(y+1)]=true; else{k[j-(y+1)]=false;} m=0;}
void L(){N(); if(n>=12) l[j-(y+1)]=true; else{l[j-(y+1)]=false;} n=0;}
void J()
```

```
    {
    if(I==iZ){J=iW;}
    else if(I==iW){J=iZ;}
    if(iI==iz) iJ=iw;
    else if(iI==iw) iJ=iz;
    }
void O(int inp,int inp0,int inp1,bool inp2,bool inp3)
    {
    if((inp<inp1)&&((Regime[inp0-(y+1)]=="sRange")||(Regime[inp0-
(y+1)]=="tRange"))){inp2=inp3;}
    else if((Regime[inp0-(y+1)]!="sRange")&&(Regime[inp0-(y+1)]!="tRange"))
inp2=!inp3; else inp2=!inp3;
    }
void R()
    {
    if(j<=J){int i=j; O=i; iO=i;}
    if((j>J)&&(j<r)){int i=j; O=i; iO=i; r=i;}
    else if(j>J){int i=j; r=i;}
    if(j<=iJ){int i=j; o=i; io=i;}
    if((j>iJ)&&(j<ir)){int i=j; o=i; io=i; ir=i;}
    else if(j>iJ){int i=j; ir=i;}
    }
bool OnHold(int inp,string inp0,string inp1){return ((Regime[inp-
(y+1)]==inp0)||(Regime[inp-(y+1)]==inp1));}
bool OnFire(int inp,string inp0,string inp1){return ((Regime[inp-
(y+1)]!=inp0)&&(Regime[inp-(y+1)]!=inp1));}
void OnPoint()
    {
    for(j=y+1;j<x; j++)
        {
        Unify(); Normalize();
        if((iStdDev<50)&&(iATR>50)) if(Regime[j-(y+1)]!="Stable"){H(); L();
if(OnFire(j,"sVolatile","tVolatile")) Regime[j-(y+1)]="sVolatile";}
        else if((iStdDev<50)&&(iATR<50))
            {
            if(Regime[j-(y+1)]!="Stable")
                {
                R(); H(); L(); if(OnFire(j,"sRange","tRange")) Regime[j-
(y+1)]="sRange";
                }
            }
        else if(OnFire(j,"sTrend","tTrend")) Regime[j-(y+1)]="sTrend";
        }
    }
void Signal()
    {
```

```
    ab=true;
    signal = price;
    }
void OnCall()
    {
    for(j=y+1;j<X+2; j++)
        {
        Normalize();
        if((Suply<=price)||(iSuply<=price)||(iSuply<=iH))
            {
            int i=j; I=iW; iZ=i; Z=i; iC=C;
            if((iw!=0)&&(jC==Cc)){h=I;} jC!=C;
            if(OnHold(j,"sTrend","tTrend")){iz=i; z=i; iI=iw; H();}
            if(X!=x-1) X++;
            }
        if((Demand>=price)||(iDemand>=price)||(iDemand>=iL))
            {
            int i=j; I=iZ; iW=i; W=i; jC=C;
            if((iz!=0)&&(iC==Cc)){h=I;} iC!=C;
            if(OnHold(j,"sTrend","tTrend")){iw=i; w=i; iI=iz; L();}
            if(X!=x-1) X++;
            }
        } X=y;
    }
void OnBar()
    {
    for(j=y+1;j<x; j++)
        {
        Unify(); Normalize();
        if((iStdDev<50)&&(iATR>50))
            {
            if(Regime[j-(y+1)]!="Stable")
                {
                if(Regime[j-(y+1)]!="tVolatile")
                    {
                    F(); H(); L(); Regime[j-(y+1)]="tVolatile";
                    }
                }
            }
        else if((iStdDev<50)&&(iATR<50))
            {
            if(Regime[j-(y+1)]!="Stable")
                {
                R(); H(); L();
                if(Regime[j-(y+1)]!="tRange")
                    {
```

```
                    F(); Regime[j-(y+1)]="tRange";
                    }
                }
            }
        else if((Regime[j-(y+1)]!="tTrend")&&(Regime[j-(y+1)]!="sTrend")&&
(LL[j-(y+1)]<Discount[j-(y+1)])&&(HH[j-(y+1)]>Premium[j-(y+1)])) Regime[j-
(y+1)]="Stable";
        else
            {
            if(Regime[j-(y+1)]!="tTrend")
                {
                F(); Regime[j-(y+1)]="tTrend";
                }
            }
        }
    if(KC==true)
        {
        if((h!=0)&&(ab==false)&&(U[O-(y+1)]=true)&&(O>2)&&(O!=x-1)/*&&
(OnFire(O,"sTrend","tTrend"))*/)
            {
            if(HH[O-(y+1)]>Premium[O-(y+1)])
                {
                h=0;
                if((C==true)&&(c==true))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,price,"O:",O,"|",C,":",c);
                    }
                else if((C==false)&&(c==false))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,price,"O:",O,"|",C,":",c);
                    }
                }
            if(LL[O-(y+1)]<Discount[O-(y+1)])
                {
                h=0;
                if((C==true)&&(c==true))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,price,"O:",O,"|",C,":",c);
                    }
                else if((C==false)&&(c==false))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,price,"O:",O,"|",C,":",c);
```

```
                    }
                }
            }
        if((h!=0)&&(ab==false)&&(U[o-(y+1)]=true)&&(o>2)&&(o!=x-1)/*&&
(OnFire(o,"sTrend","tTrend"))*/)
            {
            if(HH[o-(y+1)]>Premium[o-(y+1)])
                {
                h=o;
                if((C==false)&&(c==false))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,price,"o:",o,"|",C,":",c);
                    }
                else if((C==true)&&(c==true))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,price,"o:",o,"|",C,":",c);
                    }
                }
            if(LL[o-(y+1)]<Discount[o-(y+1)])
                {
                h=o;
                if((C==false)&&(c==false))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,price,"o:",o,"|",C,":",c);
                    }
                else if((C==true)&&(c==true))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,price,"o:",o,"|",C,":",c);
                    }
                }
            }
        }
    else
        {
        if((h!=0)&&(ab==false)&&(U[O-(y+1)]=true)&&(O>2)&&(O!=x-1)/*&&
(OnFire(O,"sTrend","tTrend"))*/)
            {
            if(HH[O-(y+1)]>Premium[O-(y+1)])
                {
                h=O;
                if((C==false)&&(c==false))
                    {
```

```
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,price,"O:",O,"|",C,":",c);
                    }
                else if((C==true)&&(c==true))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,price,"O:",O,"|",C,":",c);
                    }
                }
            if(LL[O-(y+1)]<Discount[O-(y+1)])
                {
                h=O;
                if((C==false)&&(c==false))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,price,"O:",O,"|",C,":",c);
                    }
                else if((C==true)&&(c==true))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,"O:",O,"|",C,":",c);
                    }
                }
            }
        if((h!=0)&&(ab==false)&&(U[o-(y+1)]=true)&&(o>2)&&(o!=x-1)/*&&
(OnFire(o,"sTrend","tTrend"))*/)
            {
            if(HH[o-(y+1)]>Premium[o-(y+1)])
                {
                h=o;
                if((C==true)&&(c==true))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,"o:",o,"|",C,":",c);
                    }
                else if((C==false)&&(c==false))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,"o:",o,"|",C,":",c);
                    }
                }
            if(LL[o-(y+1)]<Discount[o-(y+1)])
                {
                h=o;
                if((C==true)&&(c==true))
                    {
```

```
                    Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price,"o:",o,"|",C,":",c);
                    }
                else if((C==false)&&(c==false))
                    {
                    Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price,"o:",o,"|",C,":",c);
                    }
                }
            }
        }
    Stock=iBands(NULL,0,y,2,0,PRICE_CLOSE,MODE_UPPER,0);
    Sale=iBands(NULL,0,y,2,0,PRICE_CLOSE,MODE_LOWER,0);
    iStock=iBands(NULL,0,y,2,0,PRICE_CLOSE,MODE_UPPER,1);
    iSale=iBands(NULL,0,y,2,0,PRICE_CLOSE,MODE_LOWER,1);
    iopen=iOpen(Symbol(),0,2);
    if(signal!=0)
        {
        if(/*(price>=signal+spread)||*/(price>=signal+com))
            {
            Alert("Bought: ",price);
            }
        if(/*(price<=signal-spread)||*/(price<=signal-com))
            {
            Alert("Sold: ",price);
            }
        if(tickTock==false)
            {
            if((/*(Price>=signal)||*/(price>=iopen))&&(((iC==Cc)&&
(Price>=HH[min-(y+1)]))||((jC==Cc)&&(Price>=LL[3-(y+1)])&&(((open>=Stock)||
(Price>=Stock))||((open>=iStock)||(Price>=iStock))))))
                {
                Alert("Buy: ",price);
                if((toll==0)&&(tally=="Sell")){toll ++;} tally = "Buy";
                }
            if((/*(Price<=signal)||*/(price<=iopen))&&(((jC==Cc)&&
(Price<=LL[min-(y+1)]))||((iC==Cc)&&(Price<=HH[3-(y+1)])&&(((open<=Sale)||
(Price<=Sale))||((open<=iSale)||(Price<=iSale))))))
                {
                Alert("Sell: ",price);
                if((toll==0)&&(tally=="Buy")){toll ++;} tally = "Sell";
                }
            if((toll==1)&&(tally=="Buy"))
                {
                if((A==true)&&(v==false))
                    {
```

```
                    Alert("Bull");
                    B(); if(C==true){P();} else{Q();} GF=true;
                    } toll = 0; tally = ""; signal = 0;
                }
            if((toll==1)&&(tally=="Sell"))
                {
                if((B==true)&&(u==false))
                    {
                    Alert("Bear");
                    A(); if(C==false){P();} else{Q();} GF=true;
                    } toll = 0; tally = ""; signal = 0;
                }
            }
        }
    tickTock = false;
    }
void OnGoe()
    {
    if(ab==false)
        {
        if(KC==true)
            {
            if(((h==io)&&(z>o))||((h==iO)&&(Z>O))||((h==iz)&&(Z>z))||
((h==iZ)&&(Z<z)))
                {
                if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"|","Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"|","Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            else if(((h==io)||(h==iZ)||(h==iz)||(h==iO)))
                {
                if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
```

```
Alert("Sign.",price," h:",h,"Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            }
        else
            {
            if(((h==io)&&(z>o))||((h==iO)&&(Z>O))||((h==iz)&&(Z>z))||
((h==iZ)&&(Z<z)))
                {
                if((C==false)||(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"|","Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"|","Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            else if(((h==io)||(h==iZ)||(h==iz)||(h==iO)))
                {
                if((C==false)||(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"Z:",iZ,"z:",iz,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            }
        }
    }
void OnToe()
    {
    if(ab==false)
        {
        if(KC==true)
            {
            if(((h==io)&&(w>o))||((h==iO)&&(W>O))||((h==iw)&&(W>w))||
((h==iW)&&(W<w)))
                {
                if((C==false)&&(c==false))
```

```
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"
W<w","|","W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"
W<w","|","W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            else if(((h==io)||(h==iW)||(h==iw)||(h==iO)))
                {
                if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            }
        else
            {
            if(((h==io)&&(w>o))||((h==iO)&&(W>O))||((h==iw)&&(W>w))||
((h==iW)&&(W<w)))
                {
                if((C==false)||(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"
W<w","|","W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"
W<w","|","W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            else if(((h==io)||(h==iW)||(h==iw)||(h==iO)))
                {
```

```
                if((C==false)||(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                else
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"W:",iW,"w:",iw,"O:",iO,"o:",io,"|",C,":",c);
                    }
                }
            }
        }
    }
void OnTrack()
    {
    S=x; T=x; X=y; Y=y; datetime is=iTime(_Symbol,0,0);
    for(int s=x-1;s<S; s++)
        {
        int js=s; j=js; Normalize(); Unify();
        if((Suply<=price)||(iSuply<=price)||(iSuply<=iH))
            {
            int i=s; I=iW; j=max; Z=j; iZ=i; T++; iC=C;
            if((iw!=0)&&(jC==Cc)){h=I;} jC!=C;
            if(iStdDev>50){S++; iz=i; iI=iw; j=i; H();}
            else if(iATR<50){S++; iO=i; io=i; j=i; H();} else{j=i; H();
if(is!=t){if(OnFire(j,"Stable","tVolatile")){F(); Regime[j-
(y+1)]="tVolatile";}} else{Regime[j-(y+1)]="sVolatile";} S++;}
            }
        if((Demand>=price)||(iDemand>=price)||(iDemand>=iL))
            {
            int i=s; I=iZ; j=max; W=j; iW=i; T++; jC=C;
            if((iz!=0)&&(iC==Cc)){h=I;} iC!=C;
            if(iStdDev>50){S++; iw=i; iI=iz; j=i; L();}
            else if(iATR<50){S++; iO=i; io=i; j=i; L();} else{j=i; L();
if(is!=t){if(OnFire(j,"Stable","tVolatile")){F(); Regime[j-
(y+1)]="tVolatile";}} else{Regime[j-(y+1)]="sVolatile";} S++;}
            }
        if(s==4*max) break;
        }
    for(int s=x-1;s<S; s++)
        {
        int js=s; j=js; Normalize(); Unify();
        if((iStdDev<50)&&(iATR<50)){R(); L(); H();}
        } S=x; T=x;
    if((Z!=4*max)&&(Z>=z)){j=max-1; z=j; if(is!=t){if(Regime[j-
(y+1)]!="tTrend"){F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-
(y+1)]="sTrend";}}
    else if((Z!=4*max)&&(Z<z)){j=max; z=j; if(is!=t){if(Regime[j-
(y+1)]!="tTrend"){F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-
(y+1)]="sTrend";}} else{j=x-1; z=j; if(is!=t){if(Regime[j-(y+1)]!="tTrend")
{F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-(y+1)]="sTrend";}}
    if((W!=4*max)&&(W>=w)){j=max-1; w=j; if(is!=t){if(Regime[j-
(y+1)]!="tTrend"){F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-
(y+1)]="sTrend";}}
    else if((W!=4*max)&&(W<w)){j=max; w=j; if(is!=t){if(Regime[j-
(y+1)]!="tTrend"){F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-
(y+1)]="sTrend";}} else{j=x-1; w=j; if(is!=t){if(Regime[j-(y+1)]!="tTrend")
{F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-(y+1)]="sTrend";}}
    }
void OnStand()
    {
    S=x; T=x; X=y; Y=y; datetime is=iTime(_Symbol,0,0);
    for(int s=y+1;s>Y; s--)
        {
        if(s==1) break;
        int js=s; j=js; ir=0; ij=0; Normalize(); Unify();
        if((Suply<=price)||(iSuply<=price)||(iSuply<=iH))
            {
            int i=s; I=iW; j=min+1; Z=j; iZ=i; T--; iC=C;
            if((iw!=0)&&(jC==Cc)){h=I;} jC!=C;
            if((X!=Y)&&(iz==0)&&(iStdDev>50)){ij=i; iz=i; iI=iw; j=i; H();
if((ir==0)&&(Y!=2)){Y--;}}
            else if((X!=Y)&&(iO==0)&&(iATR<50)){iO=i; ir=i; j=i; H();
if((ij==0)&&(Y!=2)){Y--;}}
            else if(X==Y){j=i; H(); if(is!=t)
{if(OnFire(j,"Stable","tVolatile")){F(); Regime[j-(y+1)]="tVolatile";}}
else{Regime[j-(y+1)]="sVolatile";} if((Y!=2)&&(X!=2)){Y--; X--;}}
            }
        else if((Demand>=price)||(iDemand>=price)||(iDemand>=iL))
            {
            int i=s; I=iZ; j=min+1; W=j; iW=i; T--; jC=C;
            if((iz!=0)&&(iC==Cc)){h=I;} iC!=C;
            if((X!=Y)&&(iw==0)&&(iStdDev>50)){ij=i; iw=i; iI=iz; j=i; L();
if((ir==0)&&(Y!=2)){Y--;}}
            else if((X!=Y)&&(iO==0)&&(iATR<50)){iO=i; io=i; ir=0; j=i; L();
if((ij==0)&&(Y!=2)){Y--;}}
            else if(X==Y){j=i; L(); if(is!=t)
{if(OnFire(j,"Stable","tVolatile")){F(); Regime[j-(y+1)]="tVolatile";}}
else{Regime[j-(y+1)]="sVolatile";} if((Y!=2)&&(X!=2)){Y--; X--;}}
            } else{if((Y!=2)&&(X!=2)){Y--; X--;}}
        }
    for(int s=Y+1;s<y+1; s++)
        {
        int js=s; j=js; Normalize(); Unify();
        if((iStdDev<50)&&(iATR<50)){R(); L(); H();}
        } X=y; Y=y;
    if((Z!=2)&&(Z>=z)){j=min; z=j; if(is!=t){if(Regime[j-(y+1)]!="tTrend")
{F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-(y+1)]="sTrend";}}
    else if((Z!=2)&&(Z<z)){j=min+1; z=j; if(is!=t){if(Regime[j-
(y+1)]!="tTrend"){F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-
(y+1)]="sTrend";}} else{j=y+1; z=j; if(is!=t){if(Regime[j-(y+1)]!="tTrend")
{F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-(y+1)]="sTrend";}}
    if((W!=2)&&(W>=w)){j=min; w=j; if(is!=t){if(Regime[j-(y+1)]!="tTrend")
{F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-(y+1)]="sTrend";}}
    else if((W!=2)&&(W<w)){j=min+1; w=j; if(is!=t){if(Regime[j-
(y+1)]!="tTrend"){F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-
(y+1)]="sTrend";}} else{j=y+1; w=j; if(is!=t){if(Regime[j-(y+1)]!="tTrend")
{F(); Regime[j-(y+1)]="tTrend";}} else{Regime[j-(y+1)]="sTrend";}}
    }
void OnTick()
    {
    datetime is=iTime(_Symbol,0,0);
    price=SymbolInfoDouble(_Symbol,SYMBOL_BID);
    //Close
    Price=iClose(Symbol(),0,1);
    //Open
    open=iOpen(Symbol(),0,1);
    iH=iHigh(Symbol(),0,1);
    iL=iLow(Symbol(),0,1);
    if(FG==false)
        {
        ArrayResize(k,x-y);
        ArrayResize(l,x-y);
        ArrayResize(HH,x-y);
        ArrayResize(LL,x-y);
        ArrayResize(Premium,x-y);
        ArrayResize(Discount,x-y);
        ArrayResize(Regime,x-y);
        for(j=y+1;j<x; j++){F();} FG=true;
        }
    T(); OnPoint(); O(iO,O,J,C,Cc); O(io,o,iJ,c,cC); OnCall(); J();
    if(is!=t){OnBar(); O(iO,O,J,C,Cc); O(io,o,iJ,c,cC);}
    if((J==y+1)&&(J!=2))
        {
        OnStand(); J(); O(iO,O,J,C,Cc); O(io,o,iJ,c,cC);
        if((iO!=2)&&(J>=iO)){j=min; O=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        else if((iO!=2)&&(J<iO)){j=min+1; O=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}} else{j=2; O=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        if((io!=2)&&(iJ>=io)){j=min; o=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        else if((io!=2)&&(iJ<io)){j=min+1; o=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}} else{j=2; o=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        }
    if(J==x-1)
        {
        OnTrack(); J(); O(iO,O,J,C,Cc); O(io,o,iJ,c,cC);
        if((iO!=4*max)&&(J>=iO)){j=max-1; O=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        else if((iO!=4*max)&&(J<iO)){j=max; O=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}} else{j=x-1; o=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        if((io!=4*max)&&(iJ>=io)){j=max-1; o=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        else if((io!=4*max)&&(iJ<io)){j=max; o=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}} else{j=x-1; o=j; if(is!=t)
{if(OnFire(j,"Stable","tRange")){F(); Regime[j-(y+1)]="tRange";}}
else{Regime[j-(y+1)]="sRange";}}
        } t=is;
    if(Z!=x-1)
        {
        if((Z!=y+1)&&(k[iZ-(y+1)]==true)){h=iZ; OnGoe();}
        else if((k[iz-(y+1)]==true)&&(z!=y+1)&&(z!=x-1)/*&&
(OnHold(z,"tTrend","sTrend"))*/){h=iz; OnGoe();}
        else if((k[io-(y+1)]==true)&&(o!=y+1)&&(o!=x-1)/*&&
(OnHold(o,"tRange","sRange"))*/){h=io; OnGoe();}
        else if((k[iO-(y+1)]==true)&&(O!=y+1)&&(O!=x-1)/*&&
(OnHold(O,"tRange","sRange"))*/){h=iO; OnGoe();}
        }
    if(W!=x-1)
```

```
        {
        if((W!=y+1)&&(l[iW-(y+1)]==true)){h=iW; OnToe();}
        else if((l[iw-(y+1)]==true)&&(w!=y+1)&&(w!=x-1)/*&&
(OnHold(w,"tTrend","sTrend"))*/){h=iw; OnToe();}
        else if((l[io-(y+1)]==true)&&(o!=y+1)&&(o!=x-1)/*&&
(OnHold(o,"tRange","sRange"))*/){h=io; OnToe();}
        else if((l[iO-(y+1)]==true)&&(O!=y+1)&&(O!=x-1)/*&&
(OnHold(O,"tRange","sRange"))*/){h=iO; OnToe();}
        }
    if((h!=0)&&(ab==false))
        {
        if(KC==true)
            {
            if((iz>=h)&&(iz>2)&&(((iZ>2)&&((iZ==iz)||(iZ==iz+h)||
((iZ==iz+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iz)||(I==iz+h)||((I==iz+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(k[iz-(y+1)]==false)/*&&
(OnHold(z,"sTrend","tTrend"))*/)
                {
                h=iz;
                if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"iZ:",iZ,"I:",I,"|=iz:",iz,"|",C);
                    }
                else if((C==true)&&(c==true))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"iZ:",iZ,"I:",I,"|=iz:",iz,"|",C);
                    }
                }
            else if((iO>=h)&&(iO>2)&&(((iZ>2)&&((iZ==iO)||(iZ==iO+h)||
((iZ==iO+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iO)||(I==iO+h)||((I==iO+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(k[iO-(y+1)]==false)/*&&
(OnHold(O,"sRange","tRange"))*/)
                {
                h=iO;
                if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"o:",o,"iZ:",iZ,"I:",I,"|=iO:",iO,"|",C);
                    }
                if((C==true)&&(c==true))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"o:",o,"iZ:",iZ,"I:",I,"|=iO:",iO,"|",C);
                    }
                }
            if((iw>=h)&&(iw>2)&&(((iW>2)&&((iW==iw)||(iW==iw+h)||
((iW==iw+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iw)||(I==iw+h)||((I==iw+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(l[iw-(y+1)]==false)/*&&
(OnHold(w,"sTrend","tTrend"))*/)
                {
                h=iw;
                if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"iW:",iW,"I:",I,"|=iw:",iw,"|",C);
                    }
                else if((C==true)&&(c==true))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"iW:",iW,"I:",I,"|=iw:",iw,"|",C);
                    }
                }
            else if((iO>=h)&&(iO>2)&&(((iW>2)&&((iW==iO)||(iW==iO+h)||
((iW==iO+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iO)||(I==iO+h)||((I==iO+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(l[iO-(y+1)]==false)/*&&
(OnHold(O,"sRange","tRange"))*/)
                {
                h=iO;
                if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"o:",o,"iW:",iW,"I:",I,"|=iO:",iO,"|",C);
                    }
                else if((C==true)&&(c==true))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"o:",o,"iW:",iW,"I:",I,"|=iO:",iO,"|",C);
                    }
                }
            }
        else
            {
            if((iz>=h)&&(iz>2)&&(((iZ>2)&&((iZ==iz)||(iZ==iz+h)||
((iZ==iz+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iz)||(I==iz+h)||((I==iz+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(k[iz-(y+1)]==false)/*&&
(OnHold(z,"sTrend","tTrend"))*/)
            {
            h=iz;
            if((C==true)&&(c==true))
               {
               tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"iZ:",iZ,"I:",I,"|=iz:",iz,"|",C);
               }
            else if((C==false)&&(c==false))
               {
               tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"iZ:",iZ,"I:",I,"|=iz:",iz,"|",C);
               }
            }
         else if((iO>=h)&&(iO>2)&&(((iZ>2)&&((iZ==iO)||(iZ==iO+h)||
((iZ==iO+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iO)||(I==iO+h)||((I==iO+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(k[iO-(y+1)]==false)/*&&
(OnHold(O,"sRange","tRange"))*/)
            {
            h=iO;
            if((C==true)&&(c==true))
               {
               tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"o:",o,"iZ:",iZ,"I:",I,"|=iO:",iO,"|",C);
               }
            if((C==false)&&(c==false))
               {
               tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"o:",o,"iZ:",iZ,"I:",I,"|=iO:",iO,"|",C);
               }
            }
         if((iw>=h)&&(iw>2)&&(((iW>2)&&((iW==iw)||(iW==iw+h)||
((iW==iw+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iw)||(I==iw+h)||((I==iw+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(l[iw-(y+1)]==false)/*&&
(OnHold(w,"sTrend","tTrend"))*/)
            {
            h=iw;
            if((C==true)&&(c==true))
               {
               tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"iW:",iW,"I:",I,"|=iw:",iw,"|",C);
               }
            else if((C==false)&&(c==false))
               {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"iW:",iW,"I:",I,"|=iw:",iw,"|",C);
                    }
                }
            else if((iO>=h)&&(iO>2)&&(((iW>2)&&((iW==iO)||(iW==iO+h)||
((iW==iO+io)&&(l[io-(y+1)]==false)/*&&(OnHold(o,"sRange","tRange"))*/)))||
((I>2)&&((I==iO)||(I==iO+h)||((I==iO+io)&&(l[io-(y+1)]==false)/*&&
(OnHold(o,"sRange","tRange"))*/))))&&(l[iO-(y+1)]==false)/*&&
(OnHold(O,"sRange","tRange"))*/)
                {
                h=iO;
                if((C==true)&&(c==true))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sig.",price," h:",h,"o:",o,"iW:",iW,"I:",I,"|=iO:",iO,"|",C);
                    }
                else if((C==false)&&(c==false))
                    {
                    tickTock = true; Signal(); toll = 0; tally = ""; G();
Alert("Sign.",price," h:",h,"o:",o,"iW:",iW,"I:",I,"|=iO:",iO,"|",C);
                    }
                }
            }
        }
    if(GF==true){OnReInit(); GF=false;}
    Comment("    ^",iZ,":",Z,"|",iz,":",z,"=",k[Z-(y+1)],"|",k[z-(y+1)],
    "\n Lim",iO,":",O,"^",k[O-(y+1)],"_",l[O-(y+1)],".",io,":",o,"^",k[o-
(y+1)],"_",l[o-(y+1)],"=",h,".",C,":",c,
    "\n    _",iW,":",W,"|",iw,":",w,"=",l[W-(y+1)],"|",l[w-(y+1)]);
    }//U+1F48E-💎 Natalia Tanyatia
```