//+------------------------------------------------------------------+

//| Double Sma.mq4 |

//| Copyright 2017, Tom Whitbread. |

//| http://www.gript.co.uk |

//+------------------------------------------------------------------+

#property copyright "2017, Tom Whitbread."

#property link "http://www.gript.co.uk"

#property description "Smoothed Moving Average sample expert advisor"

#define MAGICNUM 20131111

// Define our Parameters

input double Lots = 0.1;

input int PeriodOne = 40; // The period for the first SMA

input int PeriodTwo = 100; // The period for the second SMA

input int TakeProfit = 40; // The take profit level (0 disable)

input int StopLoss = 0; // The default stop loss (0 disable)

//+------------------------------------------------------------------+

//| expert initialization functions |

//+------------------------------------------------------------------+

int init()

{

return(0);

}

int deinit()

{

return(0);

}

//+------------------------------------------------------------------+

//| Check for cross over of SMA |

//+------------------------------------------------------------------+

int CheckForCross(double input1, double input2)

{

static int previous\_direction = 0;

static int current\_direction = 0;

// Up Direction = 1

if(input1 > input2){

current\_direction = 1;

}

// Down Direction = 2

if(input1 < input2){

current\_direction = 2;

}

// Detect a direction change

if(current\_direction != previous\_direction){

previous\_direction = current\_direction;

return (previous\_direction);

} else {

return (0);

}

}

//+------------------------------------------------------------------+

//| Calculate optimal lot size |

//+------------------------------------------------------------------+

double LotsOptimized()

{

double lot = Lots;

// Calculate Lot size as a fifth of available free equity.

lot = NormalizeDouble((AccountFreeMargin()/5)/1000.0,1);

if(lot<0.1) lot=0.1; //Ensure the minimal amount is 0.1 lots

return(lot);

}

//+------------------------------------------------------------------+

//+ Break Even |

//+------------------------------------------------------------------+

bool BreakEven(int MN){

int Ticket;

for(int i = OrdersTotal() - 1; i >= 0; i--) {

OrderSelect(i, SELECT\_BY\_POS, MODE\_TRADES);

if(OrderSymbol() == Symbol() && OrderMagicNumber() == MN){

Ticket = OrderModify(OrderTicket(), OrderOpenPrice(), OrderOpenPrice(), OrderTakeProfit(), 0, Green);

if(Ticket < 0) Print("Error in Break Even : ", GetLastError());

break;

}

}

return(Ticket);

}

//+------------------------------------------------------------------+

//+ Run the algorithm |

//+------------------------------------------------------------------+

int start()

{

int cnt, ticket, total;

double shortSma, longSma, ShortSL, ShortTP, LongSL, LongTP;

// Parameter Sanity checking

if(PeriodTwo < PeriodOne){

Print("Please check settings, Period Two is lesser then the first period");

return(0);

}

if(Bars < PeriodTwo){

Print("Please check settings, less then the second period bars available for the long SMA");

return(0);

}

// Calculate the SMAs from the iMA indicator in MODE\_SMMA using the close price

shortSma = iMA(NULL, 0, PeriodOne, 0, MODE\_SMMA, PRICE\_CLOSE, 0);

longSma = iMA(NULL, 0, PeriodTwo, 0, MODE\_SMMA, PRICE\_CLOSE, 0);

// Check if there has been a cross on this tick from the two SMAs

int isCrossed = CheckForCross(shortSma, longSma);

// Get the current total orders

total = OrdersTotal();

// Calculate Stop Loss a