```
//@version=5
indicator('Crypto Momentum Analysis with Optional Cryptos',
shorttitle='Intermediate Momentum crypto option', overlay=false)

// Inputs for enabling and selecting the third and fourth cryptocurrencies
bool includeThirdCrypto = input(true, title='Layer 1 Crypto')
string thirdCryptoSymbol = input('SOLUSD', title='Layer 1 Crypto Symbol')
bool includeFourthCrypto = input(false, title='AI Crypto')
string fourthCryptoSymbol = input('AKTUSD', title='AI Crypto Symbol')  //
Replace 'AKTUSD' with your default fourth crypto

// Define the lookback periods and threshold
sevenDays = 7
fourteenDays = 14
twentyOneDays = 21
dailyDrawdownThreshold = -5.0

// Fetching the closing prices for each cryptocurrency
btcPrice = request.security('BTCUSD', 'D', close)
ethPrice = request.security('ETHUSD', 'D', close)
thirdCryptoPrice = includeThirdCrypto ? request.security(thirdCryptoSymbol,
'D', close) : na
fourthCryptoPrice = includeFourthCrypto ?
request.security(fourthCryptoSymbol, 'D', close) : na

// Calculating percentage change for each period for BTC and ETH
btcSevenDayChange = (btcPrice - request.security('BTCUSD', 'D',
close[sevenDays])) / request.security('BTCUSD', 'D', close[sevenDays]) * 100
ethSevenDayChange = (ethPrice - request.security('ETHUSD', 'D',
close[sevenDays])) / request.security('ETHUSD', 'D', close[sevenDays]) * 100
btcFourteenDayChange = (btcPrice - request.security('BTCUSD', 'D',
close[fourteenDays])) / request.security('BTCUSD', 'D', close[fourteenDays])
* 100
ethFourteenDayChange = (ethPrice - request.security('ETHUSD', 'D',
close[fourteenDays])) / request.security('ETHUSD', 'D', close[fourteenDays])
* 100
btcTwentyOneDayChange = (btcPrice - request.security('BTCUSD', 'D',
close[twentyOneDays])) / request.security('BTCUSD', 'D',
close[twentyOneDays]) * 100
ethTwentyOneDayChange = (ethPrice - request.security('ETHUSD', 'D',
close[twentyOneDays])) / request.security('ETHUSD', 'D',
close[twentyOneDays]) * 100

// Calculations for Third and Fourth cryptos (if included)
```

```
thirdCryptoSevenDayChange = includeThirdCrypto ? (thirdCryptoPrice -
request.security(thirdCryptoSymbol, 'D', close[sevenDays])) /
request.security(thirdCryptoSymbol, 'D', close[sevenDays]) * 100 : na
thirdCryptoFourteenDayChange = includeThirdCrypto ? (thirdCryptoPrice -
request.security(thirdCryptoSymbol, 'D', close[fourteenDays])) /
request.security(thirdCryptoSymbol, 'D', close[fourteenDays]) * 100 : na
thirdCryptoTwentyOneDayChange = includeThirdCrypto ? (thirdCryptoPrice -
request.security(thirdCryptoSymbol, 'D', close[twentyOneDays])) /
request.security(thirdCryptoSymbol, 'D', close[twentyOneDays]) * 100 : na
fourthCryptoSevenDayChange = includeFourthCrypto ? (fourthCryptoPrice -
request.security(fourthCryptoSymbol, 'D', close[sevenDays])) /
request.security(fourthCryptoSymbol, 'D', close[sevenDays]) * 100 : na
fourthCryptoFourteenDayChange = includeFourthCrypto ? (fourthCryptoPrice -
request.security(fourthCryptoSymbol, 'D', close[fourteenDays])) /
request.security(fourthCryptoSymbol, 'D', close[fourteenDays]) * 100 : na
fourthCryptoTwentyOneDayChange = includeFourthCrypto ? (fourthCryptoPrice -
request.security(fourthCryptoSymbol, 'D', close[twentyOneDays])) /
request.security(fourthCryptoSymbol, 'D', close[twentyOneDays]) * 100 : na

// Calculating daily percentage change for each cryptocurrency
btcDailyChange = (btcPrice - btcPrice[1]) / btcPrice[1] * 100
ethDailyChange = (ethPrice - ethPrice[1]) / ethPrice[1] * 100
thirdCryptoDailyChange = includeThirdCrypto ? (thirdCryptoPrice -
thirdCryptoPrice[1]) / thirdCryptoPrice[1] * 100 : na
fourthCryptoDailyChange = includeFourthCrypto ? (fourthCryptoPrice -
fourthCryptoPrice[1]) / fourthCryptoPrice[1] * 100 : na

// Averaging the results
btcAverage = (btcSevenDayChange + btcFourteenDayChange +
btcTwentyOneDayChange) / 3
ethAverage = (ethSevenDayChange + ethFourteenDayChange +
ethTwentyOneDayChange) / 3
thirdCryptoAverage = includeThirdCrypto ? (thirdCryptoSevenDayChange +
thirdCryptoFourteenDayChange + thirdCryptoTwentyOneDayChange) / 3 : na
fourthCryptoAverage = includeFourthCrypto ? (fourthCryptoSevenDayChange +
fourthCryptoFourteenDayChange + fourthCryptoTwentyOneDayChange) / 3 : na

// Initialize variables for the label and highest average
var label buyLabel = na
var string lastCrypto = ''
var float highestAvg = na

// Determine the highest average momentum or opt for USDT
string currentCrypto = ''
color currentColor = na
```

```
string labelText = ''
float currentCryptoDailyChange = na

highestAvg := btcAverage
currentCrypto := 'BTC'
currentColor := color.orange
if ethAverage > highestAvg
   highestAvg := ethAverage
   currentCrypto := 'ETH'
   currentColor := color.rgb(0, 255, 8)
if includeThirdCrypto and thirdCryptoAverage > highestAvg
   highestAvg := thirdCryptoAverage
   currentCrypto := thirdCryptoSymbol
   currentColor := color.rgb(44, 65, 254)
if includeFourthCrypto and fourthCryptoAverage > highestAvg
   highestAvg := fourthCryptoAverage
   currentCrypto := fourthCryptoSymbol
   currentColor := color.purple

// Determine daily change of the highest momentum crypto
currentCryptoDailyChange := currentCrypto == 'BTC' ? btcDailyChange :
currentCrypto == 'ETH' ? ethDailyChange : currentCrypto == thirdCryptoSymbol
? thirdCryptoDailyChange : fourthCryptoDailyChange

// Check for significant drawdown and suggest buying USDT
if currentCryptoDailyChange <= dailyDrawdownThreshold or highestAvg < 0
   currentCrypto := 'USDT'
   currentColor := color.rgb(254, 0, 0)
   labelText := 'Buy USDT Now\nSignificant Drawdown Detected'
else
   labelText := 'Buy ' + currentCrypto + ' Now\nBTC: ' +
str.tostring(btcAverage, '#.##') + '\nETH: ' + str.tostring(ethAverage,
'#.##') + (includeThirdCrypto ? '\n' + thirdCryptoSymbol + ': ' +
str.tostring(thirdCryptoAverage, '#.##') : '') + (includeFourthCrypto ? '\n'
+ fourthCryptoSymbol + ': ' + str.tostring(fourthCryptoAverage, '#.##') : '')

// Update or create the label
if na(buyLabel) or lastCrypto != currentCrypto
   if not na(buyLabel)
       label.delete(buyLabel)
   buyLabel := label.new(bar_index, highestAvg, labelText,
color=currentColor, textcolor=color.rgb(0, 0, 0),
style=label.style_label_down, size=size.small)
   lastCrypto := currentCrypto
```

```
// Plotting
plot(highestAvg, color=currentColor, title='Highest Momentum Average',
linewidth=4, style=plot.style_stepline)
hline(0, 'Baseline', color=color.black)
```