```
int start()
{
 //短期陰陽足の計算
 //短期陰陽足の終値の計算
 int limit = Bars - IndicatorCounted();  (1)

 for(int i = 0; i < limit; i++)
    {
     double Close_Sum = 0;

     for(int j = i; j < i + Average_Period_S; j++)  ア
        {
         Close_Sum += Close[j];  (2)
        }

     Close_S[i] = Close_Sum / Average_Period_S;  イ
     Close_S[i] = NormalizeDouble(Close_S[i],MarketInfo(Symbol(),MODE_DIGITS));
    }

 //短期陰陽足の始値の計算
 for(i = 0; i < limit; i++)
    {
     double Open_Sum = 0;

     for(j = i; j < i + Average_Period_S; j++)  ウ
        {
         Open_Sum += Close[j+1];  (3)
        }

     Open_S[i] = Open_Sum / Average_Period_S;  エ
     Open_S[i] = NormalizeDouble(Open_S[i],MarketInfo(Symbol(),MODE_DIGITS));
    }

 //中期陰陽足の計算
 //中期陰陽足の終値の計算
 for(i = 0; i < limit; i++)
    {
     Close_Sum = 0;

     for(j = i; j < i + Average_Period_M; j++)
        {
         Close_Sum += Close[j];
        }

     Close_M[i] = Close_Sum / Average_Period_M;
     Close_M[i] = NormalizeDouble(Close_M[i],MarketInfo(Symbol(),MODE_DIGITS));
    }

 //中期陰陽足の始値の計算  (4)
 for(i = 0; i < limit; i++)
    {
     Open_Sum = 0;

     for(j = i; j < i + Average_Period_M; j++)
        {
         Open_Sum += Close[j+1];
        }

     Open_M[i] = Open_Sum / Average_Period_M;
     Open_M[i] = NormalizeDouble(Open_M[i],MarketInfo(Symbol(),MODE_DIGITS));
    }

 //長期陰陽足の計算
 //長期陰陽足の終値の計算
 for(i = 0; i < limit; i++)
    {
     Close_Sum = 0;

     for(j = i; j < i + Average_Period_L; j++)
        {
         Close_Sum += Close[j];
        }

     Close_L[i] = Close_Sum / Average_Period_L;
     Close_L[i] = NormalizeDouble(Close_L[i],MarketInfo(Symbol(),MODE_DIGITS));
    }

 //長期陰陽足の始値の計算  (5)
 for(i = 0; i < limit; i++)
    {
     Open_Sum = 0;

     for(j = i; j < i + Average_Period_L; j++)
        {
         Open_Sum += Close[j+1];
        }

     Open_L[i] = Open_Sum / Average_Period_L;
     Open_L[i] = NormalizeDouble(Open_L[i],MarketInfo(Symbol(),MODE_DIGITS));
    }

 return(0);
}
```