Aufgabe 2:

Zeitaufwand: 30min

Über ersten String iterieren und bei jedem i einfach brutesearchpattern matching wo beim ersten string zum index j immer das i dazu gerechnet wird und falls i + j beim pattern searching out of bounds ist wird j auf 1 - i gesetzt. Abbruch bedingung ist chars nicht gleich bzw. j = i

Code:

program CyclicRotation;

procedure IsCyclicRotation(s1, s2: STRING; var pos: INTEGER; var ret: boolean);

var

  i,j,k,s1Len, s2Len: INTEGER;

begin

  s1Len := LENGTH(s1);

  s2Len := LENGTH(s2);

  ret := false;

  pos := 0;

  i := 0;

  if((s1Len = 0) and (s2Len = 0)) then

  begin

    ret := true;

    pos := 1;

  end else if(s1Len = s2Len) then

    while (i < s1Len) and (ret = false) do

    begin

      j := 1;

      k := 1;

      while((s2[i+j] = s1[k]) and (k <= s1Len)) do

      begin

        Inc(j);

        Inc(k);

        if(((i + j) > s1Len)) then

          j := 1 - i// if out of bounds set to start

        ;

      end;

      if(k = s1Len+1) then

      begin

        ret := true;

        pos := i + 1;

      end;

      Inc(i);

    end;

end;

procedure TestIsCyclicRotation(s1, s2: STRING);

var

  pos: INTEGER;

  ret: BOOLEAN;

begin

  IsCyclicRotation(s1, s2, pos, ret);

  WriteLn('Is: ''', s2, ''' a cyclic rotation of ''', s1, ''': ', ret);

  if(ret) then writeln('On Position: ', pos);

  WriteLn;

end;

begin

  // Test Case 1:

  TestIsCyclicRotation('', ''); // expected: True, On Position: 1

  // Test Case 2:

  TestIsCyclicRotation('abc', ''); // expected: False

  // Test Case 3:

  TestIsCyclicRotation('', 'abc'); // expected: False

  // Test Case 4:

  TestIsCyclicRotation('HelloWorld', 'HelloWorld'); // expected: True, On Position: 1

  // Test Case 5:

  TestIsCyclicRotation('HelloWorld', 'WorldHello'); // expected: True, On Position: 6

  // Test Case 6:

  TestIsCyclicRotation('abcd', 'abce'); // expected: False

end.

Ergebnis Tests:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung