Aufgabe 1 - (De-)Kompression von Dateien

**Lösungsidee:**Das inputfile zeilenweise lesen und je nachdem die compress oder decompress funktion auf diese zeile anwenden und die geänderte zeile dann je nachdem auf die console oder in das outputfile schreiben.

Für die compress Funktion gehe ich über alle chars der zeile drüber und falls ein char mehr als 3 mal in folge vorkommt schreib ich statt dem gegebenen die RLE schreibweise. Für decompress das gleiche nur umgekehrt.

**Zeitaufwand:** 1h 30min

**Code:**

program RLE;

uses SysUtils;

type

  OperationType = (compress, decompress);

  OutputType = (toFile, toConsole);

function CompressString(str: string): string;

var

  i, j: integer;

  currChar: char;

  compressed: string;

  // inner procedure to avoid code duplication because this has to be run inside the loop and after the loop

  procedure Update;

  begin

    if j > 2 then

      compressed := Concat(compressed, currChar, IntToStr(j))

    else if j = 2 then

      compressed := Concat(compressed, currChar, currChar)

    else

      compressed := Concat(compressed, currChar);

    j := 1;

  end;

begin

  compressed := '';

  j := 1;

  i := 1;

  currChar := str[i];

  for i := 2 to Length(str) do

    if str[i] = currChar then

      Inc(j)

    else

    begin

      Update();

      currChar := str[i];

    end;

  Update();

  CompressString := compressed;

end;

function DecompressString(str: string): string;

var

  i, j: integer;

  decompressed: string;

begin

  decompressed := '';

  i := 1;

  while i <= Length(str) do

  begin

    if (str[i] in ['0'..'9']) then

    begin

      if(i = 1) then break;

      for j := 2 to StrToInt(str[i]) do

        decompressed := Concat(decompressed, str[i-1]);

    end else decompressed := Concat(decompressed, str[i]);

    Inc(i);

  end;

  DecompressString := decompressed;

end;

procedure RunRLE(operation: OperationType; outputType: OutputType; const inFileName: string; const outFileName: string);

var

  line: STRING;

  inFile, outFile: TEXT;

begin

  Assign(inFile, inFileName);

  Reset(inFile);

  if(outputType = toFile) then

  begin

    Assign(outFile, outFileName);

    Rewrite(outFile);

  end;

  while(not Eof(inFile)) do

  begin

    ReadLn(inFile, line);

    if(operation = compress) then

      line := CompressString(line)

    else if(operation = decompress) then

      line := DecompressString(line);

    if(outputType = toFile) then

      writeln(outFile, line)

    else

      writeln(line);

  end;

  Close(inFile);

  if(outputType = toFile) then

    Close(outFile);

end;

var

  Command: string;

  InFileName, OutFileName: string;

  outType: OutputType;

begin

  if ParamCount > 0 then

    Command := ParamStr(1)

  else begin

    write('enter if you want to compress (-c) or decompress (-d) > ');

    ReadLn(Command);

  end;

  if (Command <> '-c') and (Command <> '-d') then

  begin

    WriteLn('Error: Unkowm Command - ', Command);

    writeln;

    Halt(1);

  end;

  if ParamCount > 1 then

    InFileName := ParamStr(2)

  else begin

    write('enter infilename > ');

    ReadLn(InFileName);

  end;

  if not FileExists(InFileName) then

  begin

    WriteLn('Error: input file does not exist - ', InFileName);

    writeln;

    Halt(1);

  end;

  outType := toFile;

  if ParamCount > 2 then

    OutFileName := ParamStr(3)

  else outType := toConsole;

  if ((outType = toFile) and (inFilename = outFilename)) then

  begin

    WriteLn('Error: output file can not be the same as input file - ', InFileName);

    writeln;

    Halt(1);

  end;

  if Command = '-c' then

    RunRLE(compress, outType, InFileName, OutFileName)

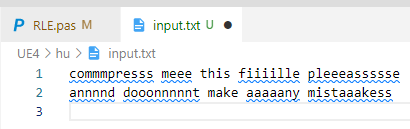
  else

    RunRLE(decompress, outType, InFileName, OutFileName);

end.

**Tests:**

Inputfile used for those tests:



**Test without any params:**

Text, letter

Description automatically generated

**Test compression with output file:**



Text

Description automatically generated

**Test decompression:**



Text

Description automatically generated

**Test Conclusion:** as we can see the compressed and again decompressed file equals the initial input.txt

Graphical user interface, website

Description automatically generated

*Src: https://text-compare.com/*