

7 Übung zu Informatik zum 10.12.2009 Blatt 7

7.1

```
program Quadrat;  
  var zahl:real;  
function quad (x:real):real; {Quadriert Gleitkommazahl}  
begin  
  quad:=x*x  
end;  
procedure eingabe; {Hauptprogramm}  
begin  
  writeln('Bitte geben Sie einen Wert x ein! p(x)=(2x^2+3x^4)^6 wird  
    berechnet!');  
  writeln('Geben Sie 0 ein, um das Programm zu beenden (p(0)=0)');  
  write('x:=');  
  readln(zahl); {Berüfung/Anweisung, Eingabe von x}  
  if not (zahl=0) then {Abbruchbedingun/Beenden}  
  begin  
    write('p(',zahl:0:2,')=',(quad(2*quad(zahl)+3*quad(zahl)*quad(zahl)) *  
      quad(quad(2*quad(zahl)+3*quad(zahl)*quad(zahl)))):0:2);  
    readln; {Ausrechnen/Anzeigen, warten auf Eingabe}  
    eingabe {Nächste Eingabe}  
  end;  
end;  
begin  
  eingabe  
end.
```

7.2

```
program GGT ;
  var zahl1:Integer; zahl2:Integer;
  function fgg(x,y:integer): integer; {Größter gemeinsamer Teiler-Funktion,
    siehe Skript Beispiel 4.1 für Alg.}
  begin
    if y = 0 then
      fgg := x {Wenn y=0, ist x unser ggT}
    else
      if x > y then {Sonst fgg mit (differenz größerem zum kleinerem) und (
        kleinerem).}
        fgg := fgg((x-y),y)
      else
        fgg := fgg(x,(y-x))
      end;
    end;
  procedure eingabe;
  begin
    writeln('Geben Sie bitte 2 Zahlen x und y ein, deren groesster
      gemeinsamer Teiler ermittelt werden soll!');
    writeln('Geben Sie fuer eine der beiden Zahlen eine 0 oder eine negative
      Zahl ein, um das Programm zu beenden');
    write('x:=');
    readln(zahl1); {Begrüßung/Anweisung, Eingabe Zahl x}
    if zahl1>0 then {Abbruchbedingung x<=0}
      begin
        write('y:=');
        readln(zahl2); {Eingabe Zahl y}
        if zahl2>0 then {Abbruchbedingung y<=0}
          begin
            writeln('Groesster gemeinsamer Teiler ist ',fgg(zahl1,zahl2)); {
              Ausgabe Ergebnis}
            eingabe {Nächste Eingabe}
          end;
        end;
      end;
    end;
  begin
    eingabe
  end.
```

7.3

```
program Romnorm;
function REPLACE(PATTERN,S,REPLACEMENT:String):String; {Ersetzen-Funktion}
var posit:integer;
begin
  posit:=pos(PATTERN,S); {Position des zu ersetzenden Teilstrings bestimmen
  .}
  if posit >0 then {Wenn Teilstring vorkommt}
  begin
    delete(S,posit,length(PATTERN)); {Alten Teilstring löschen}
    Insert(REPLACEMENT,S,posit); {Neuen Teilstring einfügen}
    posit := pos(PATTERN,S);
    if posit>0 then S:= REPLACE(PATTERN,S,REPLACEMENT); {Ersetze nochmal,
    wenn Teilstring nochmal vorkommt}
  end;
  REPLACE:=S {Bearbeiteten Text ausgeben}
end;
function NORMALIZE(Rom:String):String; {Normalisierungsfunktion}
begin
  Rom:=REPLACE('IV',Rom,'IIII'); {Ersetzt IV in Rom durch IIII}
  Rom:=REPLACE('IX',Rom,'VIIII');
  Rom:=REPLACE('XC',Rom,'LXXXX');
  Rom:=REPLACE('CM',Rom,'DCCCC');
  Rom:=REPLACE('XL',Rom,'XXXX');
  Rom:=REPLACE('CD',Rom,'CCCC');
  {Es wurden nur die angegebenen Fälle der Subtraktionsregel umgewandelt!}
  NORMALIZE:=Rom {Ausgabe normalisierte römische Zahl}
end;
procedure eingabe; {Hauptprogramm}
var romzahl:String;
begin
  writeln('Geben_Sie_eine_Roemische_Zahl_ein,_die_normailisert_werden_soll
  ');
  writeln('Lassen_Sie_den_Text_leer,_um_das_Programm_zu_beenden!');
  write('roem._Zahl:');
  readln(romzahl); {Begrüßung/Anweisung, Eingabe römische Zahl}
  if not (romzahl='') then {Abbruchbedingung keine Zahl eingegeben}
  begin
    writeln('Normalisierte_Zahl:_',Normalize(romzahl)); {Ergebnis ausgeben
    }
    eingabe {nächste Eingabe}
  end;
end;
begin
  eingabe
end.
```

7.4

```
program Romkonverter;
var dezZahl: Integer;
function REPLACE(PATTERN,S,REPLACEMENT: String): String; { Ersetzen-Funktion}
var posit: integer;
begin
  posit:=pos(PATTERN,S); {Position des zu ersetzenden Teilstrings bestimmen.}
  if posit >0 then {Wenn Teilstring vorkommt}
  begin
    delete(S,posit,length(PATTERN)); {Alten Teilstring löschen}
    Insert(REPLACEMENT,S,posit); {Neuen Teilstring einfügen}
    posit := pos(PATTERN,S);
    if posit >0 then S:= REPLACE(PATTERN,S,REPLACEMENT); {Ersetze nochmal, wenn Teilstring
      nochmal vorkommt}
  end;
  REPLACE:=S {Bearbeiteten Text ausgeben}
end;
function NORMALIZE(Rom: String): String; {Normalisierungsfunktion}
begin
  Rom:=REPLACE('IV',Rom,'IIII'); {Ersetzt IV in Rom durch IIII}
  Rom:=REPLACE('IX',Rom,'VIIII');
  Rom:=REPLACE('XC',Rom,'LXXXX');
  Rom:=REPLACE('CM',Rom,'DCCCC');
  Rom:=REPLACE('XL',Rom,'XXXX');
  Rom:=REPLACE('CD',Rom,'CCCC');
  {Es wurden nur die angegebenen Fälle der Subtraktionsregel umgewandelt!}
  NORMALIZE:=Rom {Ausgabe normalisierte römische Zahl}
end;
procedure rekumrechner(romZahl: String); {röm.->dez.}
var Ziff: String;
begin
  Ziff:=copy(romZahl,1,1); {nimm das erste Zeichen von romZahl}
  if Ziff='I' then dezZahl:=dezZahl+1; {Wenn aktuelles Zeichen '*' ist, addiere
    entsprechenden Wert zur Dezimalzahl}
  if Ziff='V' then dezZahl:=dezZahl+5;
  if Ziff='X' then dezZahl:=dezZahl+10;
  if Ziff='L' then dezZahl:=dezZahl+50;
  if Ziff='C' then dezZahl:=dezZahl+100;
  if Ziff='D' then dezZahl:=dezZahl+500;
  if Ziff='M' then dezZahl:=dezZahl+1000;
  if not (length(romZahl)=1) then {Wenn das das letzte Zeichen war, dann gehe zur Ausgabe}
  begin
    rekumrechner(copy(romZahl,2,length(romZahl)-1)) {Wenn noch Zeichen da sind, schneide
      das erste ab und mach nochmal}
  end;
end;
procedure eingabe; {Hauptprogramm}
var romzahl: String;
begin
  writeln('Geben_Sie_eine_Roemische_Zahl_ein,_die_in_eine_Dezimalzahl_umgerechnet_werden
    _soll');
  writeln('Lassen_Sie_den_Text_leer,_um_das_Programm_zu_beenden!');
  write('roemische_Zahl_Zahl:');
  readln(romzahl); {Begrüßung/Anweisung, Eingabe römische Zahl}
  if not (romzahl='') then {Abbruchbedingung keine Zahl eingegeben}
  begin
    rekumrechner(Normalize(romzahl));
    writeln('Normalisierte_Zahl:',dezZahl); {Ergebnis ausgeben}
    dezZahl:=0;
    eingabe {nächste Eingabe}
  end;
end;
begin
  eingabe
end.
```

7.5

So man die Aufgabe dementsprechend versteht, dass man 9 Streichhölzer besitzt und damit die Zahl 36 erzeugen muss, kann man den Rücken der 3 und den der 6 miteinander verbinden, um den Mangel an 2 Streichhölzchen auszugleichen. Dies sieht dann wie folgt aus: