

Klausur zur Informatik I

Justus-Liebig-Universität Gießen
Wintersemester 2000/01

Name:	13.02.2001
Vorname:	
Geburtsdatum:	
Geburtsort:	

Punktzahl:

Note:

Aufgabe 1 (10 Punkte)

Man stelle die Zahl 200,1 als 32-Bit Gleitpunktzahl in normalisierter und gerundeter (Binär-)Form dar.

Aufgabe 2 (15 Punkte)

Man erstelle einen binären AVL-Suchbaum. Dazu füge man der Reihe nach die Schlüsselwerte 42, 28, 17, 52, 69, 35 schrittweise ein und zeichne nach jeder Einfügeoperation den resultierenden Baum. Sofern Ausgleichsrotationen notwendig sind, gebe man diese und den neuen Baum ebenfalls an.

Aufgabe 3 (15 Punkte)

Im folgenden Programm befinden sich einige Fehler. Man kennzeichne diese und beschreibe kurz, was falsch ist (falsche „Korrekturen“ führen dabei zu Punktabzug).

```
PROGRAMM Fehler;  
USE crt;
```

```
VAR b,r,i = REAL;  
    n      = INTEGER;
```

```
FUNCTION Betrag(n) : INTEGER;  
BEGIN  
    IF n < 0 THEN DO  
        n := -n;  
    END;
```

```
BEGIN;  
    WRITE("Basis   : "); READLN(b);  
    WRITE("Exponent: "); READLN(n);
```

```
    r := 1,0;
```

```
    FOR  
        i := 1  
        TO bEtRaG(n) DO  
            r := r*b-c;
```

```
    IF n < 0 THEN DO  
        r := 1 / r;
```

```
    WRITLN("Ergebnis: ",r:0:5);  
END.
```

Aufgabe 4 (15 Punkte)

Man gebe eine Grammatik an, die die Sprache $\{a^n b^{2n} c^m \mid n > 0, m > 42\}$ erzeugt.

Aufgabe 5 (10 Punkte)

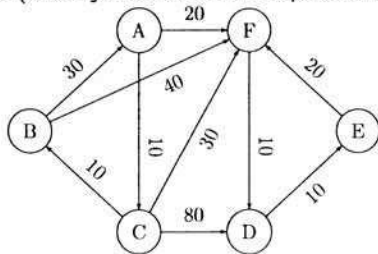
Man schreibe eine Funktion `replace`, die in einer zu übergebenden Zeichenkette alle Vokale durch einen (ebenfalls zu übergebenden) Vokal ersetzt und die neue Zeichenkette zurückgibt.

Beispielausgabe: Draa Chanasan mat dam Kantrabass

```
PROGRAM ersetzen;  
CONST s = 'Drei Chinesen mit dem Kontrabass';  
  
FUNCTION replace(...) : ... ;  
(...)  
  
BEGIN  
    WRITELN(replace(...));  
END.
```

Aufgabe 6 (15 Punkte)

Man bestimme mit dem Algorithmus von Dijkstra die kürzesten Wege von A in folgendem Graphen. (Nach jedem Schritt Graphen angeben!)



Aufgabe 7 (10 Punkte)

Welche Ausgabe bewirkt das nachfolgende Programm? Man schreibe nicht nur das Endergebnis hin, sondern begründe das Resultat.

```
PROGRAM geheim;  
VAR r,n,i : INTEGER;  
  
PROCEDURE g1(n : INTEGER);  
VAR i,r : INTEGER;  
  
BEGIN  
    r := n;  
    FOR i := 1 TO n DO  
        r := (r XOR i) MOD n;  
        n := r DIV 2;  
    END;  
  
FUNCTION g2(VAR n : INTEGER) : INTEGER;  
  
BEGIN  
    FOR i := 2 TO n DO  
        BEGIN  
            r := r + 1;  
            g1(r); g1(n);  
        END;  
        n := n div 5;  
        g2 := r;  
    END;  
  
PROCEDURE g3;  
VAR i : INTEGER;  
  
BEGIN  
    r := 3; n := r*7;  
    i := g2(n); r := i+r-n;  
END;  
  
BEGIN  
    n := 10; g1(n);  
    r := g2(n);  
    g3;  
    WRITELN(r);  
END.
```

Musterlösung zur Klausur zur Informatik I

Aufgabe 1 (10 Punkte)

Vorkommastellen: $200_{10} = 11001000_2$.

$$a_{-1} = [0, 1 \cdot 2] = [0, 2] = 0$$

$$a_{-1} = [0, 2 \cdot 2] = [0, 4] = 0$$

Nachkommastellen: $a_{-2} = [0, 4 \cdot 2] = [0, 8] = 0$

$$a_{-3} = [0, 8 \cdot 2] = [1, 6] = 1$$

$$a_{-4} = [0, 6 \cdot 2] = [1, 2] = 1$$

$$a_{-5} = [0, 2 \cdot 2] = [0, 4] = 0 \text{ (ab hier Wiederholung)}$$

Also: $0,1 = 0,00011_2$.

Insgesamt: $200,1_{10} = 11001000,00011_2$.

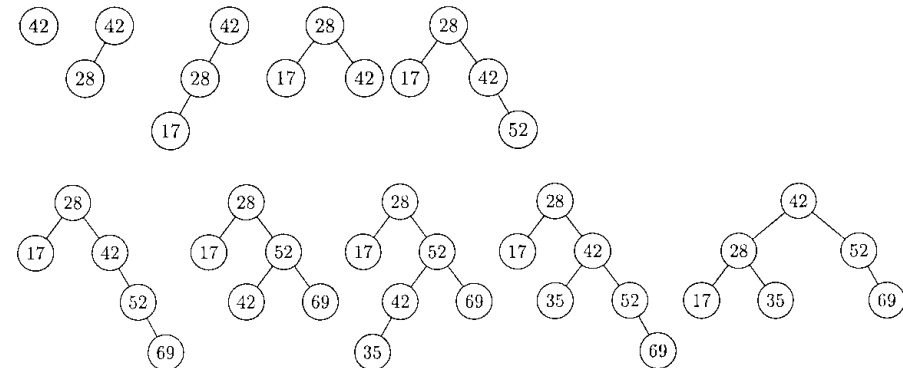
Normalisiert (und gerundet): $200,1_{10} \approx 1,10010000001100110011_2 \cdot 2^7$.

23 Bits

$\Rightarrow 200,1$ als Gleitpunktzahl:

0	10000110	10010000001100110011001
---	----------	-------------------------

Aufgabe 2 (15 Punkte)



Aufgabe 3 (15 Punkte)

PROGRAMM Fehler;
USE crt;

{ PROGRAM nur mit einem 'm' }
{ USES fehlt 's' }

VAR b,r,i = REAL;
n = INTEGER;

{ Deklaration von Variablen }
{ mit Doppelpunkt VAR b,r,i : REA

FUNCTION Betrag(n) : INTEGER;
BEGIN

{ Parameter n ohne Typ }

IF n < 0 THEN DO

{ IF n < 0 THEN ohne DO }

n := -n;

END;

{ Betrag hat keinen Rueckgabewert }

BEGIN;

WRITE("Basis : "); READLN(b); { WRITE('Basis :') nur mit }

WRITE("Exponent: "); READLN(n); { einfachen Anführungszeichen }

r := 1,0;

{ r := 1.0 Punkt statt Komma }

FOR

i := 1

{ i als REAL-Variable unzuverlässig }

TO bEtRaG(n) DO

r := r*b-c;

{ c wurde nicht deklariert }

IF n < 0 THEN DO

{ s.o. }

r := 1 / r;

WRITLN("Ergebnis: ",r:0:5);

{ WRITELN fehlt 'E' und s.o. }

END.

Aufgabe 4 (15 Punkte)

$G = (\{a, b, c\}, \{X, Y, Z\}, X, R)$ mit

$R = \{(X \rightarrow YCcc), (Y \rightarrow abb), (Y \rightarrow aYbb), (C \rightarrow Cc), (C \rightarrow c)\}$

fehlt da nicht C?

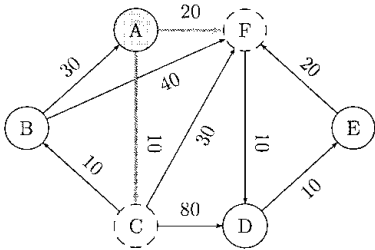
2x Kleinfunktoren "C".

Aufgabe 5 (10 Punkte)

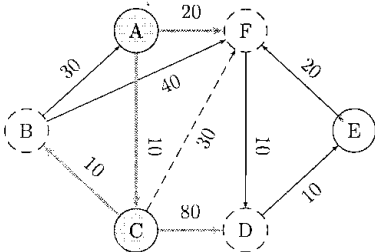
```
FUNCTION replace(text : STRING; vokal : CHAR) : STRING;  
VAR i : INTEGER;  
  
BEGIN  
  FOR i := 1 TO LENGTH(text) DO  
    IF text[i] IN ['a','e','i','o','u'] THEN  
      text[i] := vokal  
    ELSE if text[i] IN ['A','E','I','O','U'] THEN  
      text[i] := UPCASE(vokal);  
    replace := text;  
  END;
```

Aufgabe 6 (15 Punkte)

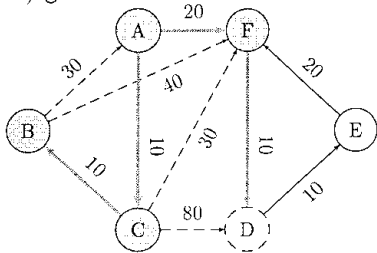
A ist gelb und wird im ersten Schritt in einen grünen Knoten umgewandelt, während C und F als seine Nachfolger gelb und die Kanten dorthin rot werden:



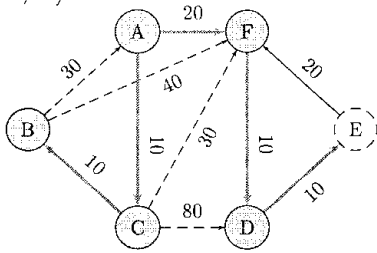
Dann wird C als gelber Knoten mit minimalem Abstand zu A grün und die noch nicht besuchten Knoten B und D gelb. Die Kante (C,F) wird gelb, da der Weg (A,F) der kürzeste ist:



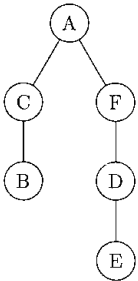
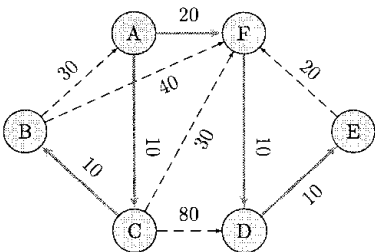
Als nächste Knoten mit minimalem Abstand zu A werden B und F grün, die Kanten (C,D) und (B,A) gelb:



D ist gelber Knoten mit minimalem Abstand zu A und wird grün und dadurch E gelb und die Kante (D,E) rot:



Im letzten Schritt wird E grüner Knoten und die Kante (E,F) gelb:



Aufgabe 7 (10 Punkte)

Die Ausgabe ist: 42.

```

PROGRAM geheim;
VAR r,n,i : INTEGER;

PROCEDURE g1(n : INTEGER);      { g1 veraendert nur lokale Variablen }
VAR i,r : INTEGER;              { kann ignoriert werden }

BEGIN
    r := n;
    FOR i := 1 TO n DO
        r := (r XOR i) MOD n;
        n := r DIV 2;
    END;

FUNCTION g2(VAR n : INTEGER) : INTEGER;

BEGIN
    FOR i := 2 TO n DO          { Variable r wird um n-1 erhoeht }
    BEGIN                        { aus g3 : r = 3, n = 21 }
        r := r + 1;             { also r = 23 }
        g1(r); g1(n);
    END;
    n := n div 5;                { aus g3 : n = 4 }
    g2 := r;                     { Rueckgabewert 23 }
END;

PROCEDURE g3;
VAR i : INTEGER;

BEGIN
    r := 3; n := r*7;           { r = 3, n = 21 }
    i := g2(n); r := i+r-n;     { i = g2(n) = 23, r = 23, n = 4 }
END;                            { => r = i+r-n = 42 !!! }

BEGIN
    n := 10; g1(n);             { alle Zuweisungen werden von g3 }
    r := g2(n);                 { wieder ueberschrieben }
    g3;                          { hier passiert etwas }
    WRITELN(r);                 { Ausgabe: 42 }
END.

```