



## THEMA II

(40 Puncte)

### 1. Es sei der nebenstehende Pseudocode-Algorithmus.

Man beschriftet mit  $[c]$  den ganzen Teil der reellen Zahl  $c$ .

- Schreibt den angezeigten Wert, wenn die Zahlen 5, 15, 27, 10, 1, 17, in dieser Reihenfolge, eingelesen werden. (6P.)
- Wenn für die Variable  $n$  der Wert 2 eingelesen wird, schreibt ein Set, von unterschiedlichen Zahlen, aus dem Intervall  $[0, 10^3]$ , die weiter eingelesen werden können, so dass nach dem Durchlaufen des Algorithmus, der Wert 4 angezeigt wird. (6P.)
- Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. (10P.)
- Schreibt in Pseudocode einen, mit dem gegebenen, äquivalenten Algorithmus, indem die Struktur **für...wiederhole** mit einer Wiederholungsstruktur vom Typ **solange...wiederhole** ersetzt wird. (6P.)

```

lese n
    (natürliche von Null
    verschiedene Zahl)
p ← 1
für i ← 1, n wiederhole
    lese x
        (natürliche Zahl)
    wiederhole
        x ← [x/3]
        bis x ≤ 3
    wenn x ≠ 0 dann
        p ← p * x
    ■
schreibe p

```

- Ein Baum mit 8 Knoten, beschriftet von 1 bis 8, wird durch den Vatervektor: (3,0,2,5,2,5,1,5) dargestellt. Bestimmt, in der Reihenfolge des Durchgangs, die Knoten der elementaren Kette mit der größten Länge und der Anfangsknoten in der Wurzel. (6P.)
- Die Variablen  $i$  und  $j$  sind vom Typ ganz, und die Variable  $a$  speichert ein zweidimensionales Feld mit 9 Zeilen und 9 Spalten, nummeriert von 0, das am Anfang alle Elemente Null hat. Schreibt die untere Anweisungssequenz und ersetzt die Auslassungspunkte mit den entsprechenden Anweisungen, davon nicht mehr als vier Zuschreibungsanweisungen, so dass nach dem Durchlaufen der erhaltenen Sequenz, die Variable  $a$  das nebenstehende Feld speichert. (6P.)

```

for (i=0; i<9; i++)
    for (j=0; j<9; j++)
        .....

```

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 |
| 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 | 2 |
| 4 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 |
| 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 |
| 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 |

## THEMA III

(30 Puncte)

- Eine natürliche Zahl heißt **major impar** wenn die Summe seiner eigenen ungeraden Teiler streng größer als die Summe seiner eigenen geraden Teiler ist. Die eigenen Teiler einer Zahl sind die natürlichen Teiler verschieden von 1 und von sich selbst.  
**Beispiel:** 18 ist eine „major impar“ Zahl (seine eigenen geraden Teiler sind 2, 6, die ungeraden 3, 9 und  $3+9 > 2+6$ ).  
Das Unterprogramm **maJImp** hat zwei Parameter,  $a$  und  $b$ , durch die es je eine natürliche Zahl ( $2 \leq a \leq b \leq 10^4$ ) erhält. Das Unterprogramm liefert die kleinste „major impar“ Zahl aus dem Intervall  $[a, b]$ , oder den Wert 0, wenn es im Intervall keine solche Zahl gibt. Schreibt in C/C++ die vollständige Definition des Unterprogramms. (10P.)  
**Beispiel:** wenn  $a=16$ ,  $b=30$ , dann liefert das Unterprogramm 18.
- In einem Text, mit höchstens 100 Zeichen, sind die Wörter gebildet aus Buchstaben des englischen Alphabets und getrennt durch je ein Leerzeichen. Der Text hat wenigstens zwei Wörter. Schreibt ein C/C++ Programm, das von der Tastatur einen Text, von dem oben angegebenen Typ, einliest und das auf dem Bildschirm die Nachricht **DA** und eine natürliche Zahl  $n$ , getrennt durch je ein Leerzeichen anschreibt, wenn alle Wörter aus dem Text je  $n$  Buchstaben haben, oder die Nachricht **NU** in dem Fall, in dem nicht alle Wörter dieselbe Anzahl von Buchstaben haben.  
**Beispiel:** wenn der eingelesene Text **Ana are cel mai bun mar** ist, wird auf dem Bildschirm **DA 3** angeschrieben und wenn der eingelesene Text **Ana are cel mai dulce mar** ist, wird auf dem Bildschirm **NU** angeschrieben. (10P.)
- Entlang einer Bergwanderungsstrecke wird eine Folge von touristischen Markierungen verwendet, die in der angegebenen Reihenfolge verfolgt werden müssen. Für jede Markierung kennt man die Quote (Höhe, gemessen in Meter) bei der es platziert ist. Man nennt **Leiter** in einer Strecke, eine Sequenz von Markierungen die sich auf aufeinanderfolgenden Positionen innerhalb der Wanderungsstrecke befinden, die als Quoten aufeinanderfolgende, streng steigend geordnete Zahlen haben. Eine Leiter ist gebildet aus wenigstens zwei Markierungen und ihre Länge ist gleich mit der Anzahl der Markierungen die sie bildet.  
Die Datei **bac.txt** enthält eine Folge von höchstens  $10^6$  natürlichen Zahlen aus dem Intervall  $[10, 10^4]$ , getrennt durch je ein Leerzeichen, die die Quoten der touristischen Markierungen aus seiner Strecke, in

der Reihenfolge in der sie erscheinen, darstellen. Schreibt auf dem Bildschirm, getrennt durch je ein Leerzeichen, in streng steigender Reihenfolge, die Quoten entsprechend einer Leiter, von maximaler Länge, dieser Strecke. Wenn es innerhalb der Strecke mehrere solche Leiter gibt, dann werden nur die Quoten der Markierungen entsprechend nur einer der Leiter angeschrieben und wenn es gar keine Leiter gibt, wird auf dem Bildschirm die Nachricht **nu exista** angeschrieben. Entwerft einen in Bezug auf die Laufzeit und den benötigten Speicher effizienten Algorithmus.

**Beispiel:** wenn die Datei die Zahlen 500 600 601 405 569 570 700 701 625 626 627 520 enthält, wird auf dem Bildschirm **625 626 627** angeschrieben.

- a. Beschreibt in Umgangssprache den verwendeten Algorithmus und begründet seine Effizienz. **(2P.)**
- b. Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. **(8P.)**