

Examenul de bacalaureat național 2018

**Proba E. d)
Informatică
Limbajul Pascal**

Varianta 2

Filiera teoretică, profilul real, specializările: matematică-informatică

matematică-informatică intensiv informatică

Filiera vocațională, profilul militar, specializarea matematică-informatică

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

THEMA I

(30 Puncte)

Für Punkt 1 schreibt auf das Prüfungsblatt den Buchstaben welcher der richtigen Antwort entspricht.

1. Die Variablen **x** und **y** sind vom Typ ganz und speichern je eine natürliche, von Null verschiedene Zahl. Ein, mit dem nebenstehenden, äquivalenter Ausdruck, kann folgender sein: **(4P.)**
- not((x mod y<>0) or (y<2))**

- a. **(x mod y=y mod x) and (y>1)** b. **((x+y) mod y=0) or (y>=1)**
c. **((x div y)*y=x) and (y>=2)** d. **(x mod y=0) or (y>2)**

2. Es sei der nebenstehende Pseudocode-Algorithmus.

Man beschriftet mit **a%b** den Rest der Teilung der natürlichen Zahl **a** durch die natürliche, von Null verschiedene, Zahl **b** und mit **[a]** den ganzen Teil **a** der reellen Zahl **a**.

- a) Schreibt die angeschriebene Zahl, wenn der Wert 2018 eingelesen wird. **(6P.)**
- b) Schreibt vier unterschiedliche Zahlen aus dem Intervall **[10,10³]**, die eingelesen werden können, so dass nach dem Durchlaufen des Algorithmus, für jede dieser der Wert 100 angeschrieben wird. **(4P.)**
- c) Schreibt im Pseudocode einen, mit dem gegebenen, äquivalenten Algorithmus, in dem die **für...wiederhole** Struktur mit einer entsprechenden Zuschreibung ersetzt wird. **(6P.)**
- d) Schreibt das dem gegebenen Algorithmus entsprechende Pascal Programm. **(10P.)**

```
lese n
    (ganze, von Null verschiedene Zahl)
wenn n<0 dann
    n←-n
■
s←0
wiederhole
    x←n%10
    für i←1,x wiederhole
        s←s+x
    ■
    n←[n/10]
bis n=0
schreibe s
```

THEMA II

(30 Puncte)

Für jeden der Punkte 1 und 2 schreibt auf das Prüfungsblatt den Buchstaben welcher der richtigen Antwort entspricht.

1. In der nebenstehenden Deklaration speichert die Variable `m`, für jedes der 20 Medikamente aus einer Apotheke, den Preis, sowie auch die Daten über die spezifische aktive Substanz: die Dosis und den Code. Ein Ausdruck dessen Wert den Code der aktiven spezifischen Substanz des ersten Medikamentes darstellt, ist: **(4P.)**

```
type medicament=record
    pret:real;
    substanta: record
        cod, doza:integer
    end
end;
var m:array[1..20] of medicament;
```

- a. `m[1].cod.substanta` b. `m[1].substanta.cod`
c. `m.cod.substanta[1]` d. `m.substanta.cod[1]`
2. In einem gerichteten Graph haben wenigstens zwei Spitzen den inneren Grad 2, wenigstens eine Spitze hat den inneren Grad 3 und wenigstens eine Spitze hat den äußeren Grad 3. Die minimale Anzahl von Spitzen des Graphs ist: **(4P.)**
- a. 3 b. 4 c. 5 d. 6

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

3. Ein Baum hat 9 Knoten, beschriftet von 1 bis 9 und die Kanten `[1,2]`, `[1,6]`, `[1,8]`, `[1,9]`, `[2,3]`, `[2,7]`, `[4,5]`, `[5,7]`. Schreibt drei Knoten auf, die als Wurzel ausgewählt werden können, so dass der Knoten 2 eine minimale Anzahl von Nachfolger haben soll. **(6P.)**

4. Die Variablen `i` und `j` sind vom Typ ganz und die Variable `a` speichert ein zweidimensionales Feld mit 9 Reihen und 9 Spalten, beschriftet von 0 bis 8, die anfangs alle Elemente gleich -1 haben. Ohne andere Variablen zu benützen, schreibt die untere Anweisungssequenz, indem ihr die Auslassungspunkte ersetzt, so dass nach dem Durchführen der erhaltenen Sequenz, die Variable `a` das nebenstehende Feld speichern soll.

```
for i:=0 to 8 do
    for j:=0 to 8 do
        .....
```

(6P.)

0	1	2	3	4	5	6	7	0
1	0	3	4	5	6	7	0	1
2	3	0	5	6	7	0	1	2
3	4	5	0	7	0	1	2	3
4	5	6	7	0	1	2	3	4
5	6	7	0	1	0	3	4	5
6	7	0	1	2	3	0	5	6
7	0	1	2	3	4	5	0	7
0	1	2	3	4	5	6	7	0

5. Gegeben wird ein Wort `s`, gebildet nur aus Buchstaben und ein Code `c`, derselben Länge wie `s`, gebildet nur aus Ziffern. Man nennt **Kodifizierung** von `s` anhand des Code `c` die Operation die eine neue Folge baut, in der Anfangs der erste Buchstabe aus `s` kopiert wird, nachher, die restliche Folge `s`, von links nach rechts, durchgehend, wird der kurrente Buchstabe am Anfang der neuen Folge eingefügt, wenn die Ziffer von derselben Position in `c`, eine Paarziffer ist, oder am Ende der neuen Folge, im Gegenfall.

Beispiel: wenn `s` die Folge, etalon und der Code 025843 ist, erhält man das Wort oltean (Anfangs enthält die Folge den Buchstaben e, nachher werden die Buchstaben t, l und o am Anfang, in der Reihenfolge in der man `s` durchgeht, eingefügt und die restlichen Buchstaben am Ende).

Schreibt ein C/C++ Programm, dass zwei Wörter, beschriftet mit `s` und `c`, von der Tastatur einliest, jedes Wort mit höchstens 10^2 Zeichen. `s` ist nur aus Kleinbuchstaben des englischen Alphabetes und `c` nur aus Ziffern gebaut. Nach dem ersten Wort wird Enter eingetippt. Das Programm baut im Speicher und schreibt auf dem Bildschirm, das Wort erhalten durch die Kodifizierung von `s` anhand von `c`, wenn die zwei Wörter dieselbe Länge haben, oder die Nachricht `cod incorrect`, im Gegenfall.

Beispiel: wenn die nebenstehenden Wörter eingelesen werden, wird auf dem Bildschirm das Wort `oltean` angeschrieben.

`etalon`
`025843`

(10P.)

