

Examenul de bacalaureat național 2017
Proba E. d)
Informatică
Limbajul Pascal

Varianta 7

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. Der nebenstehende Pascal Ausdruck hat den Wert: **(4P.)** | $3+7 \text{ div } 4+3$
- a. 1 b. 5.5 c. 7 d. 7.75

2. 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 $[c]$ den ganzen Teil der reellen Zahl c .

- a) Schreibt was angeschrieben wird, wenn in dieser Reihenfolge die Zahlen 4, 5, 24, 3, 45 gelesen werden.

(6P.)

- b) Wenn die erste gelesene Zahl 3 ist, schreibt einen Set von unterschiedlichen Zahlen, welche nachfolglich gelesen werden können, so dass nach dem Durchlaufen des Algorithmus der Wert 1 angezeigt wird. **(4P.)**

- c) Schreibt in Pseudocode einen, mit dem gegebenen, äquivalenten Algorithmus, in dem ihr die erste Struktur, **solange...wiederhole** mit einer Wiederholungsstruktur vom Typ **für...wiederhole** ersetzt. **(6P.)**

```
lese n
(natürliche von Null verschiedene
Zahl)
k ← 0
i ← 1
solange i ≤ n wiederhole
| lese x
| (natürliche von Null verschiedene
Zahl)
| y ← 2
| solange x > 1 und x % y ≠ 0
wiederhole
| | y ← y + 1
| | ■
| | wenn k < [x/y] dann
| | | k ← [x/y]
| | ■
| i ← i + 1
| ■
schreibe k
```

- d) Schreibt das dem gegebenen Algorithmus entsprechende Pascal Programm.

(10P.)

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 sind in den Feldern **s** und **d** des Datensatzes (Verbundes) die linke, beziehungsweise die rechte Extremität jeder der 20 geschlossenen Intervalls gespeichert. Ein Pascal Ausdruck dessen Wert die linke Extremität des ersten Intervalls ist, ist: (4P.)


```
type interval=record
    s,d:integer
end;
var v:array[1..20] of interval;
```

a. **v[1].s** b. **s.v[1]** c. **v(1).s** d. **s(1).v**
2. In einem Baum mit 50 Knoten, nummeriert von 1 bis 50, ist die Wurzel der Knoten 1, und der Vater jedwelchen Knotens **i** ist der Knoten nummeriert mit $\lfloor i/2 \rfloor$. Die Länge der Kette mit der einen Extremität im Knoten 14 und der anderen Extremität im Knoten 47 ist: (4P.)

a. 5 b. 8 c. 16 d. 33

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

3. Ein ungerichteter Graph mit 4 Knoten nummeriert von 1 bis 4, ist durch die nebenstehende Adjazenzmatrix dargestellt. Schreibt die Menge der Knoten und Kanten eines Untergraphen verschieden von sich selber in dem es wenigstens einen Knoten mit dem Grad 1 und wenigstens einen Knoten mit dem Grad 2 gibt. (6P.)

0	1	0	1
1	0	0	1
0	0	0	1
1	1	1	0
4. In der nebenstehenden Struktur sind die Variablen **n** und **i** vom Typ ganz und die Variable **s** erlaubt das Speichern eines Wortes, eine Folge von höchstens 20 Zeichen. Die gelesenen Wörter sind aus Kleinbuchstaben des englischen Alphabetes gebildet und sind durch Enter getrennt.

Ohne andere Variable zu benutzen, schreibt eine Sequenz indem ihr die Auslassungspunkte, ersetzt, so dass nach dem Durchlaufen der erhaltenen Sequenz die Variable **n** die Anzahl der gelesenen Wörter für die die Unterfolge aus deren letzten drei Zeichen gebildet mit der Folge **are** übereinstimmt und diese erscheint **NICHT** mehr auf einer anderen Stelle im Wort.

Beispiel: wenn die nebenstehenden Wörter gelesen werden hat die Variable **n** den Wert 3.

```
n:=.....;
for i:=1 to 11 do
begin
    readln(s);
    .....
end;
```

la
proiectare
se
are
in
vedere
evitarea
deformatiilor
areolare
si
liniare

(6P.)
5. Schreibt ein Pascal Programm, das von der Tastatur eine natürliche gerade Zahl **n** ($n \in [2, 50]$) liest und die Elemente eines zweidimensionalen Feldes mit **n** Zeilen und **n** Spalten mit reellen Zahlen, danach das Feld im Speicher umwandelt, indem eine neue Zeile in seine Mitte eingefügt wird. Der Wert jedes Elementes ist gleich mit dem arithmetischen Mittel der Elemente die sich auf der entsprechenden Spalte des gelesenen Feldes befindet. Das erhaltene Feld zeigt auf dem Bildschirm je eine Zeile des Feldes auf je eine Zeile des Bildschirms an und die Elemente jeder Zeile werden durch je ein Leerzeichen getrennt.

Beispiel: für **n=4** und das Feld

1.5	0.75	3	1.2		erhält man das	1.5	0.75	3	1.2
2.75	1.25	2.25	1.5		Feld:	2.75	1.25	2.25	1.5
4.5	1.15	3	4.5			2.25	1	2.5	2.55
0.25	0.85	1.75	3			4.5	1.15	3	4.5
						0.25	0.85	1.75	3

(10P.)

THEMA III

(30 Puncte)

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

1. Das Unterprogramm **f** ist nebenstehend definiert. Der Wert von **f(1234)** ist:

(4P.)

```
function f(n:integer):integer;  
begin  
  if n<>0 then  
    f:=(n mod 2)*(n mod 10)+f(n div 10)  
  else  
    f:=0  
  end;  
end;
```

a. 0

b. 4

c. 6

d. 10

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen

2. Seien gegeben eine endliche Menge und einen Paaren Set der Form (i, j) , wobei i und j Elemente der Menge sind. Man nennt **zur Menge geordnete topologisch sortierte Folge**, eine Folge gebildet aus alle Elementen der Menge, so dass für jedwelches Paar (i, j) aus den gegebenen, das Element i vor dem Element j in der Folge steht..

Die Backtracking Methode verwendend, alle topologisch sortierten Folge der Menge $\{1, 2, 3, 4, 5, 6\}$ erzeugt, hinsichtlich auf die Paare $(4, 6)$, $(3, 4)$, $(3, 1)$, $(2, 5)$. Die ersten drei erzeugten Lösungen sind in dieser Reihenfolge: $(2, 3, 1, 4, 5, 6)$, $(2, 3, 1, 4, 6, 5)$, $(2, 3, 1, 5, 4, 6)$. Schreibt die vierte und die fünfte Lösung, in der Reihenfolge ihrer Erzeugung. **(6P.)**

3. Das Unterprogramm **triple** hat drei Parametern, a , b und c , durch das es je eine natürliche Zahl aus dem Intervall $[0, 10^4]$ ($a < b$) bekommt. Das Unterprogramm zeigt auf dem Bildschirm alle Triplets von natürlichen Zahlen x , y und z an, Lösungen der Gleichung $x+y+z=c$, mit der Eigenschaft, dass $a \leq x \leq y \leq z \leq b$. Jedes Triplett ist zwischen geschlungene Klammern geschrieben und die Zahlen x , y und z sind in dieser Reihenfolge, durch je ein Komma getrennt angezeigt. Wenn es kein Triplett mit der angeforderten Eigenschaft gib, wird die Nachricht **nu exista** angezeigt.

Schreibe die vollständige Definiton des Unterprogramms.

Beispiel: wenn $a=2$, $b=4$ und $c=8$, werden auf dem Bildschirm die Triplets

$\{2, 2, 4\}$ $\{2, 3, 3\}$ angezeigt

und wenn $a=5$, $b=8$, $c=8$ wird auf dem Bildschirm die Nachricht

nu exista angezeigt. **(10P.)**

4. Eine Folge ist eine **arithmetische Folge mit der Ratio r** , wenn jedwelches ihrer Glieder, außer dem ersten Glied, durch die Addition der Zahl r zu der Zahl, die in der Folge vor ihr steht, erhalten wird.

Beispiel: die Folge 12, 14, 16, 18, 20 ist eine arithmetische Folge mit Ratio 2.

Die Datei **bac.in** enthält eine Folge von höchstens 10^6 natürliche Zahlen aus dem Intervall $[0, 10^3]$, getrennt durch je ein Leerzeichen.

Prüft ob eine natürliche Zahl r existiert, so dass alle **unterschiedlichen** Zahlen der Folge so umgeordnet werden können, dass sie eine arithmetische Folge mit Ratio r bilden. Es wird auf dem Bildschirm die Zahl r , oder die Nachricht **nu**, wenn es keine solche Zahl gibt, angezeigt. Benützt einen, in Bezug auf die Laufzeit, effizienten Algorithmus.

Beispiel: wenn die Datei die Zahlen

180 30 80 280 130 330 230 30 30 330 80 enthält

Wird auf dem Bildschirm

50 angezeigt

a) Beschreibt in Umgangssprache den benötigten Algorithmus und begründet seine Effizienz. **(2P.)**

b) Schreibt das dem geschriebenen Algorithmus entsprechende Pascal Programm. **(8P.)**