

Examenul de bacalaureat național 2013
Proba E. d)
Informatică
Limbajul C/C++

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 Variable **x** ist vom Typ ganz und kann eine natürliche Zahl mit höchstens zwei Ziffern speichern. Den maximalen Wert, den der nebenstehende **C/C++** Ausdruck haben kann, ist: **x%4**
(4P.)
- a. **3** b. **24.75** c. **95** d. **396**

2. Sei der nebenstehende Pseudocode-Algorithmus.

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

- a) Schreibt den angeschriebenen Wert, wenn man, in dieser Reihenfolge, die Zahlen **19**, **23** und **2** einliest. **(6P.)**
- b) Wenn für die Variable **a** der Wert **1** und für die Variable **c** der Wert **1** eingelesen wird, schreibt alle natürlichen Zahlen die eingelesen werden können für die Variable **b**, so dass nach dem Durchführen des Algorithmus, für jede dieser der Wert **12** angeschrieben wird. **(4P.)**

lese **a, b, c** (natürliche Zahlen,
a ≤ b, 0 ≤ c ≤ 9)

```
s ← 0
für x ← a, b wiederhole
  y ← x
  solange y > 0 wiederhole
    wenn y % 10 = c dann
      s ← s + 1
    y ← [y / 10]
schreibe s
```

- c) Schreibt in Pseudocode einen mit dem gegebenen, äquivalenten Algorithmus, in dem man die Struktur **solange...wiederhole** mit einer fußgesteuerten Wiederholungsstruktur ersetzen soll. **(6P.)**
- d) Schreibt das dem gegebenen Algorithmus entsprechende **C/C++** 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. Der Ausdruck `strlen("bine")` hat den Wert: **(4P.)**
a. 1 b. 4 c. 5 d. 6
2. Sei der gerichtete Graph mit 6 Spitzen und ohne Kreisläufe. Die maximale Anzahl von Bögen des Graphs ist: **(4P.)**
a. 5 b. 7 c. 10 d. 15

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

3. Sei die nebenstehende Deklaration. Schreibt eine Sequenz von Anweisungen nach deren Durchführung das Feld `paritate` der Variablen `x` den Buchstaben `p` speichern soll, wenn die Zahl gespeichert in dem entsprechenden Feld `v` gerade ist oder den Buchstaben `i` wenn die Zahl gespeichert in dem entsprechenden Feld `v` ungerade ist. **(6P.)**
- ```
struct numar
{ int v;
 char paritate;
} x;
```
4. Sei ein Baum mit 5 Knoten, aus denen nur drei den Grad gleich mit 1 haben. Schreibt zwei Werte die die Grade der anderen zwei Knoten darstellen. **(6P.)**
5. Schreibt ein C/C++ Programm, das von der Tastatur zwei natürliche Zahlen `m` und `n` ( $2 \leq m \leq 50$ ,  $2 \leq n \leq 50$ ) einliest und die Elemente eines bidimensionalen Feldes mit `m` Reihen und `n` Spalten, natürliche Zahlen mit höchstens vier Ziffern jede. Das Programm bestimmt die Spalten, die alle Elemente gleich haben mit demselben Wert und für jede solche Spalte wird der betreffende Wert auf dem Bildschirm angeschrieben. Die angeschriebenen Werte sind getrennt durch je ein Leerzeichen und wenn es keine solchen Werte gibt, schreibt das Programm die Nachricht **Nu exista**.

**Beispiel:** für `m=4`, `n=5` und das nebenstehenden Feld werden auf dem Bildschirm die Zahlen 3 7 7, nicht unbedingt in dieser Reihenfolge, angeschrieben.

**(10P.)**

|   |   |   |   |   |
|---|---|---|---|---|
| 2 | 3 | 7 | 5 | 7 |
| 3 | 3 | 7 | 4 | 7 |
| 7 | 3 | 7 | 1 | 7 |
| 8 | 3 | 7 | 6 | 7 |

**THEMA III**

**(30 Puncte)**

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

1. Seien die Unterprogramme **f1** und **f2**, unterstehend definiert.

```
int f1 (int x, int y)
{
 return x*y;
}
```

```
int f2 (int x, int y)
{ if (y==1) return x;
 else return x*f2(x, y-1);
}
```

Bei dem Aufruf, wird für die Parameter **x=2** und **y=3**, **x<sup>y</sup>** geliefert: **(4P.)**

- a. sowohl **f1**, als auch **f2**      b. nur **f1**      c. nur **f2**      d. nicht **f1**, nicht **f2**

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

2. Die Backtracking Methode benützend erzeugt, alle Wörter mit **5** Buchstaben aus der Menge **{A, M}**, Wörter die mit **M** beginnen und enden. Die ersten vier erzeugten Lösungen, in dieser Reihenfolge, sind: **MAAAM**, **MAAMM**, **MAMAM**, **MAMMM**. Schreibt die fünfte und die sechste Lösung, in der Reihenfolge in der sie erzeugt wurden, an. **(6P.)**

3. Sei das Unterprogramm **valuri**, mit zwei Parameter:
- **n**, durch den es einen natürlichen Wert **2 < n < 50** bekommt;
  - **v**, durch den es ein eindimensionales Feld mit **2 · n** Elementen liefert, ganze Werte aus dem Intervall **[1, 2 · n]**.

Das Unterprogramm baut das Feld **v**, so dass in diesem, die Folge der ungeraden Elemente streng steigend sein soll und die Folge der geraden Elemente streng fallend sein soll. Das erste Element der Folge ist ungerade und zwei Elemente mit derselben Parität können nicht aufeinanderfolgende Positionen in dem Feld besetzen, wie im Beispiel.

Schreibt die vollständige Definition des Unterprogramms.

**Beispiel:** wenn **n=4** dann, nach dem Aufruf, **v=( 1, 8, 3, 6, 5, 4, 7, 2 )**. **(10P.)**

4. Sei die nebenstehend definierte Folge (wobei **n** eine natürliche, von Null verschiedene Zahl ist), in der es keine zwei Glieder gibt die dieselbe Parität auf aufeinanderfolgenden Positionen haben:
- $$f_n = \begin{cases} 1, & \text{wenn } n = 1 \\ 1 + f_{n-1}, & \text{wenn } n \text{ gerade} \\ 1 + 2 \cdot f_{n-2}, & \text{sonst} \end{cases}$$
- 1, 2, 3, 4, 7, 8, 15, 16 . . . .**

Es wird von der Tastatur eine natürliche Zahl **x** eingelesen, mit höchstens neun Ziffern, Glied der gegebenen Folge, und man soll in der Textdatei **bac.txt**, in streng fallender Reihenfolge, getrennt durch je ein Leerzeichen, alle Glieder der Folge schreiben, die kleiner oder gleich mit **x** sind.

Für das Bestimmen der verlangten Glieder benützt man einen im Bezug auf den benötigten Speicher und der Laufzeit effizienten Algorithmus.

**Beispiel:** wenn **x=15**, enthält die Datei **bac.txt** die Zahlen

**15 8 7 4 3 2 1**

- a) Beschreibt in Umgangssprache den benützten Algorithmus, indem ihr seine Effizienz erklärt. **(4P.)**

- b) Schreibt das dem beschriebenen Algorithmus entsprechende **C/C++** Programm. **(6P.)**