

Examenul național de bacalaureat 2023

Proba E. d)
Informatică
Limbajul C/C++

Varianta 7

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei ore.
- Identificatorii utilizați în rezolvări 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). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

THEMA I

(20 Puncte)

Für Punkt 1 bis 5 schreibt auf das Prüfungsblatt den Buchstaben welcher der richtigen Antwort entspricht. Jede richtige Antwort wird mit 4 Punkten bewertet.

1. Bestimmt die Zahlen, die die ganzen Variablen **x** und **y** speichern können, so dass der Wert des nebenstehenden C/C++ Ausdrucks 1 sein soll.
- $y \% x - (x / y) * 3 \neq 0$**
- a. **x=25** und **y=75** b. **x=25** und **y=15** c. **x=15** und **y=0** d. **x=10** und **y=30**
2. Das Unterprogramm **f** ist nebenstehend definiert. Bestimmt den Wert **f(3, 2)**.
- ```
int f(int x, int y)
{
 int z;
 if (y==0) return 1;
 z=f(x,y/2);
 if (y%2!=0) return z*z*x;
 return z*z;
}
```
- a. 1      b. 2      c. 9      d. 18
3. Durch die Variable **s** hat man Zugang zu einer Folge von höchstens 20 Zeichen. Bestimmt die Folge **s** nach dem Durchlaufen der nebenstehenden Sequenz.
- ```
strcpy(s, "2019");
strcpy(s+3, "23");
```
- a. 20123 b. 201923 c. 202223 d. 2023
4. Unter Verwendung der Backtracking Methode werden alle Parfüms, gebildet durch die Mischung von je 3 unterschiedlichen Essenzen aus der Menge {**bergamotă**, **cireș**, **iris**, **lămâie**, **salcâm**} erzeugt. Zwei Parfüms sind unterschiedlich, wenn sie wenigstens eine Essenz verschieden haben. Die ersten vier Lösungen sind, in dieser Reihenfolge: (**bergamotă**, **cireș**, **iris**), (**bergamotă**, **cireș**, **lămâie**), (**bergamotă**, **cireș**, **salcâm**) und (**bergamotă**, **iris**, **lămâie**). Bestimmt die vorletzte erzeugte Lösung.
- a. (**cireș**, **iris**, **salcâm**) b. (**cireș**, **lămâie**, **salcâm**)
c. (**lămâie**, **iris**, **salcâm**) d. (**iris**, **lămâie**, **salcâm**)
5. Ein ungerichteter Graph hat 5 Knoten, nummeriert von 1 bis 5 und mit den Graden in der nebenstehenden Tabelle angezeigt. Bestimmt ein Paar von möglichen Werten für **x** und **y**.
- | | | | | | |
|--------|---|----------|---|---|----------|
| Knoten | 1 | 2 | 3 | 4 | 5 |
| Grad | 2 | x | 3 | 3 | y |
- a. 0 und 4 b. 1 und 5 c. 2 und 3 d. 3 und 3

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 3, 746, 82, 3067, 67, 78, 178, in dieser Reihenfolge, gelesen werden. (6P.)
- Wenn für n die Zahl 2 gelesen wird, schreibt eine Folge von natürlichen Zahlen im Intervall $[0, 9]$, die danach so in der Reihenfolge gelesen werden können, dass nach dem Durchlaufen des Algorithmus für jede dieser der Wert 0 angezeigt wird. (6P.)
- Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. (10P.)
- Schreibt in Pseudocode einen, mit dem gegebenen, äquivalenten Algorithmus, in dem die erste Struktur **solange...wiederhole** mit einer Wiederholungsstruktur vom Typ **für...wiederhole** ersetzt wird. (6P.)

lese n (eine von Null verschiedene natürliche Zahl)

```
x ← 0
solange n > 0 wiederhole
    lese a, b (natürliche Zahlen)
    c ← a; p ← 1
    solange a > 9 wiederhole
        a ← [a/10]; p ← p * 10
    a ← a * p + b
    wenn a ≠ c dann
        x ← x + 1
    n ← n - 1
schreibe x
```

- In einem Baum mit Wurzel befindet sich ein Knoten auf dem Niveau x , wenn die elementare Kette ein Ende in dem entsprechenden Knoten und das andere Ende in der Wurzel des Baums hat und ihre Länge ist x . Auf dem Niveau 0 befindet sich ein einziger Knoten (die Wurzel). Ein Baum mit 6 Knoten, nummeriert von 1 bis 6, hat die Kanten $[1, 2]$, $[2, 3]$, $[2, 6]$, $[3, 4]$, $[3, 5]$. Schreibt die Knoten, die als Wurzel bezeichnet werden können, so dass jeder der erhaltenen Bäume eine minimale Anzahl von Niveaus hat. (6P.)
- Die Variable p , nebenstehend deklariert, speichert die Eigenschaften eines Produkts: Name und Preis. Schreibt die C/C++ Anweisungssequenz durch die die Variable a den Wert des ersten Buchstaben des betreffenden Produktnamens speichert, wenn dieses den Preis streng kleiner als 100 hat, oder anders das Zeichen *. (6P.)

```
struct Produkt
{
    char Name[20];
    int Preis;
};
char a;
```

THEMA III

(30 Puncte)

- Das Unterprogramm **DNPI** hat einen einzigen Parameter n , durch das es eine natürliche Zahl ($n \in [1, 10^9]$) erhält und zeigt auf dem Bildschirm, durch je ein Leerzeichen getrennt, alle positiven, ungeraden Teiler von n an, die **KEINE** Primzahlen sind. Schreibt die vollständige Definition des Unterprogramms.
Beispiel: wenn $n=90$, so werden auf dem Bildschirm, nicht unbedingt in dieser Reihenfolge, die Zahlen 1 9 15 45 angezeigt. (10P.)
- Ein Spiel verwendet eine rechteckige Tafel, auf der gleich große Zellen auf m Reihen und n Spalten dargestellt sind. In jede Zelle ist eine natürliche Zahl eingeschrieben. Man nennt **Quadrat mit dem Wert p** vier Zellen der Tafel, die sich auf zwei aufeinanderfolgenden Reihen und auf zwei aufeinanderfolgenden Spalten befinden, mit der Eigenschaft, dass die Summe der eingeschriebenen Werte gleich mit p ist. Der Zweck des Spiels ist die Bestimmung eines Quadrats von maximalem Wert.
Schreibt ein C/C++ Programm, das von der Tastatur zwei natürliche Zahlen aus dem Intervall $[2, 20]$ m und n , danach die Elemente eines zweidimensionalen Feldes mit m Reihen und n Spalten, natürliche Zahlen des Intervalls $[0, 10^4]$, einliest. Diese stellen die eingeschriebenen Zahlen auf der Spieltafel, in der Reihenfolge der entsprechenden Zellen, dar. Das Programm bestimmt das Quadrat mit dem maximalen Wert der Tafel und schreibt auf dem Bildschirm diesen Wert an.
Beispiel: für $m=5$, $n=4$ und das nebenstehende Feld, wird auf dem Bildschirm der Wert 36, entsprechend dem in der Abbildung hervorgehobenen Quadrats, angeschrieben. (10P.)
- Die Datei **date.in** enthält auf der ersten Reihe zwei natürliche Zahlen des Intervalls $[1, 10^6]$, m und n , und auf den nächsten zwei Reihen natürliche Zahlen des Intervalls $[0, 10^2]$: also, auf der zweiten Reihe eine Folge **A** von m Zahlen und auf der dritten Reihe eine Folge **B** von n Zahlen. Die Zahlen auf derselben Reihe sind durch je ein Leerzeichen getrennt. Schreibt auf dem Bildschirm die maximale Anzahl der Paare der Form (p_a, p_b) ($p_a \in [1, m]$, $p_b \in [1, n]$) an, mit der Eigenschaft, dass das Glied auf der Position p_a , entsprechend der Folge **A**, denselben Wert mit dem Glied auf der Position p_b der Folge **B** hat und jede Position, entsprechend der Folge **A**, beziehungsweise der Folge **B**, in höchstens

1	1	2	30
3	10	1	2
1	13	12	1
1	2	3	1
7	1	20	1

einem Paar, wie in dem Beispiel, erscheint. Erstellt einen in Bezug auf die Laufzeit effizienten Algorithmus.

Beispiel: wenn die Datei die nebenstehenden Zahlen enthält, wird auf dem Bildschirm 6 angeschrieben.

(zum Beispiel, für die Paare (1, 1), (2, 9), (4, 2), (5, 5), (6, 6), (7, 7)
oder für die Paare (1, 2), (2, 9), (4, 1), (5, 7), (6, 8), (8, 5)).

8	9						
<u>1</u>	<u>0</u>	4	<u>1</u>	<u>5</u>	<u>3</u>	<u>5</u>	5
<u>1</u>	<u>1</u>	1	7	<u>5</u>	<u>3</u>	<u>5</u>	<u>3</u>
							<u>0</u>

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