

Examenul de bacalaureat național 2017
Proba E. d)
Informatică
Limbajul C/C++

Simulare

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 nebenstehenden C/C++ Ausdruck hat den Wert: **(4P.)** | 17/3/2%17
- a. 0 b. 2 c. 10 d. 17

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 .

- a) Schreibt was angeschrieben wird, wenn die eingegebenen Werte, in dieser Reihenfolge 15, 3 und 4 sind. **(6P.)**
- b) Schreibt zwei unterschiedliche Eingabe-Datensets, so dass nach dem Durchlaufen des Algorithmus, für jede von diesen, der Wert 0 angezeigt wird. **(4P.)**
- c) Schreibt in Pseudocode, einen mit dem gegebenen äquivalenten Algorithmus, in dem ihr die Struktur **solange ... wiederhole** mit einer Wiederholungsstruktur für...wiederhole ersetzt. **(6P.)**

```
lese n,a,b
(natürliche, von Null verschiedene Zahlen,
 $a \leq n, b \leq n$ )
ok ← 0
x ← 1
solange  $x \leq n$  wiederhole
| wenn  $x \% a = 0$  und  $x \% b \neq 0$  oder
|    $x \% a \neq 0$  und  $x \% b = 0$  dann
|   schreibe x, ' '
|   ok ← 1
|   ■
|   x ← x+1
|   ■
wenn ok=0 dann
|   schreibe 0
|   ■
```

- 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. In der folgenden Sequenz speichert die Variable *s* eine Folge mit höchstens 20 Zeichen.

```
strcpy(s,"tezauriza");  
cout<<strstr(s,"za"); | printf("%s",strstr(s,"za"));
```

Nach dem Durchlaufen der Sequenz, wird auf dem Bildschirm folgendes angezeigt: **(4P.)**

- a. 3 b. 2 c. za d. zauriza
2. Die Adjazenzmatrix eines ungerichteten Graphen mit 7 Knoten hat 10 von Null verschiedene Elemente. Die maximale Anzahl von zusammenhängenden Komponenten des Graphen ist: **(4P.)**
- a. 2 b. 3 c. 4 d. 5

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

3. Sei ein Baum mit 8 Knoten, beschriftet von 1 bis 8, mit dem Wurzel 5 und die Kanten [1,5], [2,7], [3,7], [3,8], [4,5], [5,7], [6,7]. Zählt die Knoten auf, die, die direkte Nachfolger (Söhne) des Knoten 7 sind. **(6P.)**

4. Die Variable *m* speichert gleichzeitig, für jeden der 20 Autos die zu vermieten sind, die folgenden Daten: Baujahr des Autos (natürliche Zahl) und Typ des Brennstoffes des Autos (Folge von höchstens 50 Zeichen).

Wenn man weist, dass die unteren C/C++ Ausdrücke das Baujahr und Brennstofftyp des dritten Autos speichern, schreibt die Definition einer Struktur mit Etikett *masina*, die, die Daten eines Autos speichert und vereinbare die entsprechende Variable *m*.

m[2].an

m[2].carburant

(6P.)

5. Schreibt ein C/C++ Programm das von der Tastatur eine natürliche Zahl *n* ($n \in [2, 20]$) liest und im Speicher ein zweidimensionales Feld mit *n* Zeilen und *n* Spalten baut in dem:
- die letzte Spalte enthält die natürlichen Zahlen aus dem Intervall [1, *n*], in streng fallender Reihenfolge;
 - Alle Elemente der ersten Zeile haben den Wert *n*;
 - Jedes andere Element wird als Summe der mit ihm benachbarten Elemente erhalten, ein Element an der Spalte nach rechts und auf derselben Zeile, das andere Element an derselben Spalte und auf der vorigen Zeile, wie im Beispiel.

Das Programm zeigt auf dem Bildschirm das so erhaltene Feld an, jede Zeile des Feldes an eine Zeile des Bildschirms, die Elemente derselben Zeile durch je ein Leerzeichen getrennt.

Beispiel: für *n*=4 wird an Bildschirm:

```
4   4   4   4  
15  11  7   3  
35  20  9   2  
65  30  10  1
```

angezeigt.

(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. Gibt an, was nach dem Durchlaufen des Aufrufs `f(6)` angezeigt wird;

(4P.)

```
void f(int x)
{ cout<<x; | printf("%d",x);
  if(x>3)
  { f(x-2);
    cout<<x; | printf("%d",x);
  }
}
```

- a. 642 b. 6446 c. 64246 d. 642246

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

2. Es wird die Backtracking Methode verwendet um alle Blumensträucher aus drei unterschiedlichen Frühlingsblumen aus der Menge {`brândușă`, `iasomie`, `lalea`, `liliac`, `margaretă`} zu erzeugen, so dass Die Jasmine (`iasomia`) und die Flieder (`liliacul`) werden nicht in demselben Strauch eingefügt. Wenn man weiß, dass in einem Strauch die Reihenfolge der Blumen unwichtig ist, sind die ersten vier erzeugten Lösungen, in dieser Reihenfolge: (`brândușă`, `iasomie`, `lalea`), (`brândușă`, `iasomie`, `margaretă`), (`brândușă`, `lalea`, `liliac`), (`brândușă`, `lalea`, `margaretă`). Schreibt die fünfte und die sechste Lösung, in der Reihenfolge ihrer Erzeugung. **(6P.)**

3. Das Unterprogramm `ranguri` hat einen einzigen Parameter, `n`, durch den es eine natürliche Zahl bekommt ($n \in [0, 10^9]$). Das Unterprogramm liefert die Anzahl der Ziffern von `n`, die denselben Wert haben, wie die Stelle die sie besetzen, in der Schreibweise der Zahl. Die Stellen werden von rechts nach links nummeriert, die Einheitsziffer besitzt die Stelle 0.

Schreibt die vollständige Definition des Unterprogramms.

Beispiel: wenn `n=6594270`, das Unterprogramm liefert die Zahl 4.

(10P.)

4. Die Datei `bac.in` enthält in der ersten Zeile zwei natürliche Zahlen aus dem Intervall $[2, 10^4]$, `m` und `n`, und auf jeden der folgenden zwei Zeilen je eine Folge aus `m`, beziehungsweise `n` natürliche Zahlen aus dem Intervall $[0, 10^9]$, **streng steigend** geordnet. Die Zahlen derselben Zeile sind durch je ein Leerzeichen getrennt.

Es werden auf dem Bildschirm, in **streng fallender Reihenfolge**, die Paarzahlen, die mindestens in einer der zwei Folgen vorkommen. Die angeforderten Zahlen werden, durch je ein Leerzeichen getrennt, angezeigt, wenn es keine solche Zahl gibt, wird auf dem Bildschirm die Nachricht `nu exista` angezeigt.

Für das Bestimmen der angeforderten Zahlen benutzt einen in Bezug auf die Laufzeit effizienten Algorithmus.

Beispiel: Wenn die Datei folgende Zahlen enthält

5 6

1 4 8 9 10

2 4 10 12 15 18

wird auf dem Bildschirm

18 12 10 8 4 2

angezeigt.

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

- b) Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. **(8P.)**