

Examenul de bacalaureat 2011

Proba E. d)

Proba scrisă la INFORMATICĂ

Limbajul C/C++

Varianta 3

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 efectiv de lucru 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).

THEMA I

(30 Puncte)

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

1. Der C/C++ Ausdruck

`!(-20 >= x) && (x <= -10) || (x >= -5) && !(x > 5)`

hat Wert 1 wenn und nur wenn der Wert gespeichert in der reellen Variablen **x** der Vereinigung der unteren Intervalle gehört. **(4P.)**

a. $(-\infty, -10] \cup [5, \infty)$

b. $[-20, -10] \cup (-5, 5)$

c. $(-20, -10) \cup (-5, 5)$

d. $(-20, -10] \cup [-5, 5]$

2. Sei nebenstehender Pseudocode-Algorithmus.

Man bezeichnet mit $x \% y$ den Rest der Division der natürlichen Zahl **x** durch die natürliche von Null verschiedene Zahl **y** und mit $[z]$ den ganzen Teil der reellen Zahl **z**.

a. Schreibt die Zahl, die angeschrieben wird, nach dem Durchführen des Algorithmus wenn für die Variable **n** der Wert 1034 eingelesen wird und die Variable **m** Wert 1234 hat. **(6P.)**

b. Schreibt alle unterschiedlichen Paare von natürlichen, zweistelligen Zahlen die für die Variablen **n** und **m** eingelesen werden können, so dass nach dem Durchführen des Algorithmus, für jede dieser Zahlen 86 angeschrieben wird. **(6P.)**

c. Schreibt einen äquivalenten Pseudocode-Algorithmus der keine Wiederholungsstrukturen benutzen soll. **(4P.)**

d. Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. **(10P.)**

```
lies n, m
    (natürliche Zahlen)
wenn n < m dann
    x ← n
    n ← m
    m ← x
■
p ← 1
solange m > 0 wiederhole
    c ← m % 10
    m ← [m / 10]
    n ← n - p * c
    p ← p * 10
■
schreibe n
```

THEMA II

(30 Puncte)

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

1. Sei der gerichtete Graph mit 6 Spitzen, nummeriert von 1 bis 6, dargestellt durch die nebenstehenden Adjazenzlisten. Zwei Wege sind unterschiedlich wenn sie sich durch wenigstens einen Bogen unterscheiden.
Die Anzahl der unterschiedlichen elementaren Wege von der Spitze 2 zu der Spitze 3 sind: **(4P.)**

| | |
|----|-------------|
| 1: | 3 |
| 2: | 1, 5, 6 |
| 3: | leere Liste |
| 4: | 3 |
| 5: | leere Liste |
| 6: | 4, 5 |

a. 0 b. 1 c. 2 d. 3
2. Die Blätter des Baumes mit Wurzel, mit 8 Knoten, nummeriert von 1 bis 8, dargestellt durch den "Vatervektor" (6, 6, 5, 0, 6, 4, 4, 7) ist: **(4P.)**

a. 1, 2, 3, 8 b. 1, 2, 8 c. 3, 7 d. 4, 6, 7

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

3. Der strukturierte Datentyp **CARTE** erlaubt das Speichern einer reellen Zahl, die die Summe in Euro eines Buches darstellt, einer ganzen Zahl, die die Anzahl der Seiten des Buches darstellt und eine Zeichenfolge die den Titel des Buches darstellt, gebildet aus höchstens 50 Zeichen.
Schreibt in der C/C++ Sprache eine Definition des gegebenen Datentyps und deklariert eine Variable **x** von diesem Typ. Benennt ausdrucksvoll die Felder dieses Datentyps. **(6P.)**
4. In der nebenstehenden Programmsequenz speichert die Variable **a** eine Folge von höchstens 100 Zeichen und die Variablen **i** und **k** sind ganz.
Schreibt die nach dem Durchführen der Sequenz, auf dem Bildschirm, angeschriebene Folge. **(6P.)**

```

k='a'-'A';
strcpy(a,"bacalaureat");
for(i=0;i<strlen(a);i++)
    if(strchr("aeiou",a[i])!=NULL)
        a[i]=a[i]-k;
cout<<a; | printf("%s",a);

```
5. Schreibt ein C/C++ Programm, das von der Tastatur zwei natürliche Zahlen, **n** und **m** ($2 < n \leq 24$, $2 < m \leq 24$), einliest und im Speicher ein bidimensionales Feld mit **n** Reihen und **m** Spalten baut, in welches jedes Element das sich auf der letzten Reihe oder auf der letzten Spalte befindet den Wert 1 hat und jedwelches andere Element gleich ist mit der letzten Ziffer der Summe der beiden benachbarten Elementen, die sich auf derselben Reihe befinden aber auf der Spalte nach rechts, bzw. auf derselben Spalte aber auf der nächsten Reihe.
Das Programm schreibt auf dem Bildschirm das erhaltene Feld, je eine Reihe des Feldes auf je einer Bildschirmreihe, die Elemente jeder Reihe sind getrennt durch je ein Leerzeichen. **(10P.)**

Beispiel: für **n=4** und **m=5** das nebenstehende Feld angeschrieben.

| | | | | |
|---|---|---|---|---|
| 5 | 0 | 0 | 4 | 1 |
| 5 | 0 | 6 | 3 | 1 |
| 5 | 4 | 3 | 2 | 1 |
| 1 | 1 | 1 | 1 | 1 |

THEMA III

(30 Puncte)

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

1. Man erzeugt durch die Backtracking Methode in steigender Reihenfolge alle natürlichen Zahlen von je vier Ziffern aus der Menge $A=\{1,2,3,4,5\}$, Zahlen die keine zwei nebeneinanderliegenden ungeraden Ziffern enthalten. Die ersten acht erzeugten Zahlen, in dieser Reihenfolge, sind: 1212, 1214, 1221, 1222, 1223, 1224, 1225, 1232. Die Anzahl der erzeugten Werte die die Tausenderziffer gleich mit 2 und die Einheitsziffer gleich mit 4 haben, ist: **(4P.)**
- a. 20 b. 16 c. 12 d. 9

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

2. Schreibt was angeschrieben wird, nach dem unterstehenden Aufruf, des nebenstehenden Unterprogramms `f`. `f(12345)`; **(6P.)**
- ```
void f (int x)
{ cout<<"*"; | printf("*");
 if(x>0) {
 cout<<x; | printf("%d",x);
 f(x/100);
 cout<<"*"; | printf("*");
 }
}
```
3. Das Unterprogramm `inter` hat vier Parameter:
- `na`, `nb` durch die es je eine natürliche Zahl bekommt ( $0 < na < 100$ ,  $0 < nb < 100$ );
  - `a`, `b` durch die es je ein eindimensionales Feld bekommt, das eine Menge von `na`, bzw. `nb` natürlichen Zahlen speichert, jede mit höchstens vier Ziffern.
- Das Unterprogramm liefert die Anzahl der Elemente die dem Schnitt der beiden Mengen gehört.
- Schreibt in C/C++ Sprache die vollständige Definition des Unterprogramms `inter`.
- Beispiel:** für `na=4`, `a=(35, 149, 72, 3798)`, `nb=5`, `b=(72, 151, 149, 9, 623)`, liefert das Unterprogramm, nach dem Aufruf, den Wert 2. **(10P.)**
4. Die Datei `BAC.TXT` enthält eine Folge von mindestens 11 und höchstens eine Million natürlichen Zahlen, getrennt durch je ein Leerzeichen. Jede Zahl hat mindestens zwei und höchstens neun Ziffern. Das erste Bestandteil der Folge hat die Ordnungszahl 1, das zweite Bestandteil hat die Ordnungszahl 2 usw.
- Liest die Folge aus der Datei. Bestimmt und schreibt am Bildschirm, die Ordnungszahl eines Bestandteils der Folge, welches die maximale Anzahl von Werte hat, die mit ihm die gleiche Zehnerziffer haben und vor ihn in der Datei vorkommen. Man soll einen im Bezug auf die Laufzeit effizienten Algorithmus verwenden. Wenn es mehrere Bestandteile gibt mit dieser Eigenschaft schreibt man die Ordnungszahl nur eines Bestandteils.
- Beispiel:** wenn die Datei `BAC.TXT` die Zahlen  
12 36 265 18 139 19 32 34 112 14 68 enthält  
wird auf dem Bildschirm 10 angeschrieben (die Ordnungszahl des Bestandteils 14).
- a) Beschreibt in der Umgangssprache den benötigten Algorithmus und erklärt worin seine Effizienz besteht. **(4P.)**
- b) Schreibt das C/C++ Programm entsprechend dem beschriebenen Algorithmus. **(6P.)**