

**Examenul de bacalaureat 2012**  
**Proba E. d)**  
**Proba scrisă la INFORMATICĂ**  
**Limbajul C/C++**

**Varianta 1**

**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).
- 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. Bestimmt den Ausdruck der den Wert 1 hat, wenn und nur wenn die Zahl gespeichert in der ganzen Variablen  $x$  der Menge  $\{1, 2, 3\}$  angehört. **(4P.)**

- a.  $x==1 \ \&\& \ x==2 \ \&\& \ x==3$   
c.  $x==1 \ || \ x==2 \ || \ x==3$

- b.  $!( \ x<=1 \ \&\& \ x>=3 \ )$   
d.  $!( \ x<=1 \ || \ x>=3 \ )$

**2. Sei der nebenstehende 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 angezeigt wird, wenn für die Variable  $n$  der Wert 56413 gelesen wird. **(6P.)**  
b) Schreibt alle natürlichen Zahlen, jede mit **genau** vier Ziffern, die für die Variable  $n$  eingelesen werden können, so dass nach dem Durchführen des Algorithmus, für jede dieser Zahlen der Wert 40 angezeigt wird. **(4P.)**

```
lese n (natürliche Zahl)
m←0
p←1
solange n>0 wiederhole
| wenn n%2≠0 dann
| | n←n-1
| | ■
| m←m+(n%10)*p
| n←[n/10]
| p←p*10
| ■
schreibe m
```

- c) Schreibt im Pseudocode einen Algorithmus, äquivalent mit dem gegebenen, in dem die Struktur **solange...wiederhole** mit einer Wiederholungsstruktur von einem anderen Typ ersetzt wird. **(6P.)**  
d) Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. **(10P.)**

**THEMA II**

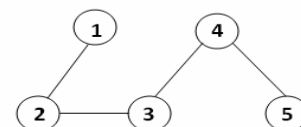
**(30 Puncte)**

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

1. Sei der gerichtete Graph mit 7 Knoten, nummeriert von 1 bis 7, und die Bögen  $(1,2)$ ,  $(2,3)$ ,  $(3,4)$ ,  $(4,1)$ ,  $(5,4)$ ,  $(7,5)$ ,  $(5,6)$ . Die Anzahl der Knoten des Graphen mit gleichen inneren und äußeren Grad, ist: **(4P.)**

a. 2                      b. 3                      c. 4                      d. 5

2. Sei der ungerichtete Graph mit fünf Knoten, nebenstehend dargestellt. Die minimale Anzahl von Kanten die dem Graph hinzugefügt werden können, so dass es zwischen jedwelchen zwei unterschiedlichen Knoten wenigstens eine elementare Kette der Länge 2 gibt, ist: **(4P.)**



a. 1                      b. 2                      c. 3                      d. 4

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

3. Seien die folgenden Deklarationen. Die Variable **s** speichert die Koordinaten, im **xOy** Koordinatensystem, der Endpunkte einer Strecke. Schreibt eine Anweisung die am Bildschirm die Quadratlänge der Strecke, die der Variablen **s** entspricht, anzeigt. **(6P.)**

```
struct punct {
    float x,y;
};
struct segment {
    punct A,B;
}s;
```

4. In der unteren Anwesungssequenz sind die Variablen **i** und **j** vom Typ ganz, und die Variable **A** speichert ein zweidimensionales Feld mit 5 Zeilen und 5 Spalten, nummeriert von 1 bis 5. Die Elemente des Feldes sind ganze Zahlen. Ohne andere Variablen zu verwenden, schreibt eine, oder mehrere Anweisungen, welche die Auslassungspunkte ersetzen können, so dass nach dem Durchführen der enthaltenen Sequenz, das in der Variable **A** gespeicherte Feld die Elemente aus der unteren Figur enthalten soll.

```
for i:=1 to 5 do
    for j:=1 to 5 do
        .....
```

**(6P.)**

```
2 3 4 0 1
3 4 0 1 2
4 0 1 2 3
0 1 2 3 4
1 2 3 4 0
```

5. Seien zwei Zeichenfolgen **a**, mit der Länge **na** und **b**, mit der Länge **nb**. Die Kette **a** ist **prefix** der Kette **b** wenn  $na \leq nb$  und die Unterkette von **b**, bestimmt von den ersten **na** Zeichen, identisch mit **a** ist. Die Kette **a** ist **suffix** der Kette **b**, wenn  $na \leq nb$  und die Unterkette von **b**, bestimmt von den letzten **na** Zeichen, identisch mit **a** ist.

Schreibt ein C/C++-Programm das von der Tastatur zwei Wörter einliest, jedes mit höchstens 30 Zeichen, nur Kleinbuchstaben des englischen Alphabets. Nach dem ersten Wort wird Enter getippt. Das Programm zeigt am Bildschirm, getrennt durch je ein Leerzeichen, alle Sufixe des ersten Wortes die Prefixe des zweiten Wortes sind wie im Beispiel. Wenn es keine solche Unterkette gibt wird am Bildschirm die Nachricht **NU EXISTA** angezeigt.

**Beispiele:**

wenn, in dieser Reihenfolge, die Wörter:

**rebele**

**elegant** eingelesen werden

werden die Unterketten: **e ele**, (nicht unbedingt in dieser Reihenfolge) angezeigt

wenn, in dieser Reihenfolge, die Wörter

**strai**

**rai** eingelesen werden,

wird die Unterkette **rai** angezeigt

**(10P.)**

**THEMA III**

**(30 Puncte)**

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

1. Die Backtracking Methode verwendend werden in lexikographischer Reihenfolge, alle Folgen von 5 Ziffern der Menge  $\{0, 1, 2\}$ , erzeugt, Folgen mit der Eigenschaft, dass auf den aufeinanderfolgenden Stellen keine zwei geraden Ziffern sind. Die ersten acht erzeugten Folgen, in dieser Reihenfolge, sind: 01010, 01011, 01012, 01101, 01110, 01111, 01112, 01121. Die neunte erzeugte Folge ist: **(4P.)**

- a. 01120                      b. 01201                      c. 01210                      d. 10101

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

2. Sei das Unterprogramm  $f$ ,  
nebenstehend definiert. Schreibt was  
nach dem Aufrufen von  $f(9)$   
angezeigt wird; **(6P.)**

```
void f(int x)
{ cout<<x; | printf("%d",x);
  if(x>0)
  { if(x%2==0)
    cout<<' '; | printf("");
    f(x/2);
  }
}
```

3. Sei das Unterprogramm `double`, mit zwei Parameter:
- $n$ , durch das es einen natürlichen Wert  $2 < n < 50$  bekommt;
  - $v$ , durch das es ein eindimensionales Feld mit  $n$  Elementen, mit höchstens 4-stelligen natürlichen Zahlen bekommt. Wenigstens ein Element des Feldes ist streng positiv.
- Das Unterprogramm fügt nach jedem streng positiven Element des Feldes ein neues Element, mit demselben Wert ein, wie im Beispiel. Das veränderte Feld, so wie der aktualisierte Wert von  $n$ , wird durch die Parameter  $v$ , beziehungsweise  $n$  geliefert.
- Schreibt die vollständige Definition des Unterprogramms.
- Beispiel:** wenn  $n=6$  und  $v=(4, -5, 0, 9, 9, -2)$ , dann ist nach dem Aufruf  $n=9$ , und  $v=(4, 4, -5, 0, 9, 9, 9, 9, -2)$ . **(10P.)**

4. Eine wenigstens zweistellige natürliche Zahl heißt **x-ordonat** wenn alle ihre Ziffern in steigender Reihenfolge sind und der Absolutbetrag der Differenz jedwelcher ihrer zwei aufeinanderfolgenden Ziffern gleich mit  $x$  ist.
- Beispiel :** die Zahl 2468 ist 2-ordonat, die Zahl 147 ist 3-ordonat; die Zahlen 179 oder 131 sind nicht vom dem beschriebenen Typ.
- Es wird von Tastatur eine natürliche Zahl  $x$  ( $1 \leq x \leq 8$ ) gelesen. Schreibt in die Datei **BAC.TXT** alle unterschiedlichen natürlichen Zahlen die **x-ordonate** sind, je eine Zahl auf eine Zeile der Datei.
- Für das Bestimmen der verlangten Zahlen wird ein Algorithmus verwendet der im Bezug mit der Laufzeit effizient ist.
- a) Beschreibt in der Umgangssprache den Algorithmus und erklärt worin seine Effizienz besteht. **(4P.)**
- b) Schreibt das dem beschriebenen Algorithmus entsprechende C/C++ Programm. **(6P.)**