

**Examenul de bacalaureat 2011**  
**Proba E. d)**  
**Proba scrisă la INFORMATICĂ**  
**Limbajul Pascal**

**Varianta 9**

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 Lösung entspricht.**

1. Gebt an welche der unterstehenden **Pascal** Ausdrücke den Wert **true** haben, wenn und nur wenn die Zahl, die in der ganzen Variable **x** gespeichert ist, der Vereinigung der Intervalle  $[-3, -1] \cup [1, 3]$  gehört. **(4P.)**

- a.  **$x \geq -3 \ \&\& \ x \leq -1 \ \&\& \ x \geq 1 \ \&\& \ x \leq 3$**
- b.  **$!(x < -3 \ || \ x > -1) \ || \ !(x < 1 \ || \ x > 3)$**
- c.  **$x \geq -3 \ || \ x \leq -1 \ || \ x \geq 1 \ || \ x \leq 3$**
- d.  **$!(x < -3 \ \&\& \ x > 3 \ \&\& \ x > -1 \ || \ x < 1)$**

**2. Sei nebenstehender Pseudocode-Algorithmus.**

- a. Schreibt, in Ordnung, die Zahlen die nach der Durchführung des Algorithmus angeschrieben werden, wenn für **n** der Wert 5 und für **k** der Wert 2 eingelesen wird. **(6P.)**
- b. Wenn für **k** der Wert 5 gelesen wird, schreibt den kleinsten und den größten Wert, der für die Variable **n** gelesen werden kann, so dass nach der Durchführung des Algorithmus, für jeden Wert die letzte Zahl die angeschrieben wird 7 ist. **(6P.)**
- c. Schreibt einen Pseudocode-Algorithmus, der nur eine statt zwei Wiederholungsstrukturen enthält und äquivalent mit dem gegebenen ist. **(4P.)**
- d. Schreibt das dem gegebenen Algorithmus entsprechende **Pascal** Programm. **(10P.)**

```
lies n,k
    (natürliche, von Null verschiedene
     Zahlen)
t ← 0
solange n ≥ 1 wiederhole
    wenn n > k dann i ← k
    sonst i ← n
    t ← t + 1
    n ← n - i
solange i ≥ 1 wiederhole
    schreibe t, ' '
    i ← i - 1
```

**(30 Punkte)**

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

1. Man nennt Wald einen ungerichteten Graph in dem jedes konnexe Bestandteil ein Baum ist. Jedwelcher Wald mit wenigstens zwei Bäume ist ein Graph der: (4P.)

a. Zyklen hat und konnex ist	b. Zyklen hat und nicht konnex ist
c. keine Zyklen hat und konnex ist	d. keine Zyklen hat und nicht konnex ist
2. Sei der gerichtete Graph mit 6 Spitzen, nummeriert von 1 bis 6 und die Menge der Bögen  $\{(1, 4), (2, 1), (2, 4), (2, 6), (4, 1), (4, 6)\}$ . Die Anzahl der Untergraphen, wobei jede die Menge der Bögen  $\{(1, 4), (2, 1), (2, 4), (4, 1)\}$  hat, ist: (4P.)

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

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

3. Sei der ungerichtete Graph mit 5 Knoten, nummeriert von 1 bis 5, dargestellt durch die nebenstehenden Adjazenzlisten.  
Zählt die Knoten auf die zu dem konnexen Bestandteil des gegebenen Graphs gehören und das die wenigsten Knoten hat. (6P.)

1:	4, 5
2:	3
3:	2
4:	1
5:	1

4. In der unteren Anweisungssequenz sind die Variablen **i** und **j** vom Typ ganz und die Variable **A** ist ein bidimensionales Feld mit 5 Reihen und 5 Spalten, nummeriert von 0 bis 4, mit den Elementen ganze Zahlen.  
Ohne andere Variablen zu benützen, schreibt eine Anweisung die die Auslassungspunkte ersetzen kann, so dass nach dem Durchführen der erhaltenen Sequenz, das Feld gespeichert in der Variablen **A** die Elemente aus der unterstehenden Figur haben soll. Vor dem Durchführen der Sequenz sind alle Elemente des Feldes Null.

for i:=0 to 4 do	0 1 2 3 4
for j:=0 to 4 do	1 2 3 4 5
.....	2 3 4 5 6
	3 4 5 6 7
	4 5 6 7 8

5. Schreibt ein **Pascal** Programm, das von der Tastatur eine natürliche Zahl **n** ( $2 \leq n \leq 20$ ) einliest und nachher **n** unterschiedliche Wörter, jedes gebildet aus höchstens 20 Zeichen, nur Kleinbuchstaben des englischen Alphabets. Bei der Dateneingabe, gibt man nach jedem Wort Enter ein. Das Programm schreibt auf dem Bildschirm die Anzahl der Wörter aus den letzten **n-1** eingelesenen, die mit dem ersten eingelesenen Wort beginnen.  
**Beispiel:** wenn **n=5** und die eingelesenen Wörter:  
**bun**  
**buncar**  
**bunici**  
**abundent**  
**bunavoie**  
sind, wird auf dem Bildschirm 3 angeschrieben (weil nur die Wörter **buncar**, **bunici** und **bunavoie** mit **bun** beginnen). (10P.)

**TEMA III**

**(30 Puncte)**

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

1. Seien die unterstehenden rekursiven Unterprogramme **c1** und **c2**.

```
function C1 (a,b:integer):integer;  
begin  
  if a=b then C1:=a  
  else if a>b then C1:=C1(a-b,b)  
    else C1:=C1(a,b-a)  
end;
```

```
function C2  
(a,b:integer):integer;  
begin  
  if b=0 then C2:=a  
  else C2:=C2(b,a mod b)  
end;
```

Nach dem Aufruf, liefert den Wert des größten gemeinsamen Teilers der beiden natürlichen, von Null verschiedenen Zahlen, als Parameter zugeschickt: **(4P.)**

- a. nur **c1**                      b. nur **c2**                      c. sowohl **c1** als auch **c2**                      d. auch nicht **c1**, auch nicht **c2**

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

2. Fünf Perlen von verschiedenen Farben **roșu**, **galben**, **verde**, **albastru**, **violet** zur Verfügung habend, benutzt man die Backtracking Methode um alle Möglichkeiten zu erhalten, durch denen man Ketten von je drei Perlen bilden kann. Man weiß dass innerhalb einer Kette die Reihenfolge der Perlen wichtig ist. Die Farben der Perlen entsprechend den ersten vier erhaltenen Lösungen sind, in dieser Reihenfolge: (**roșu**, **galben**, **verde**), (**roșu**, **galben**, **albastru**), (**roșu**, **galben**, **violet**), (**roșu**, **verde**, **galben**). Schreibt die letzten zwei Lösungen, in der Reihenfolge in der sie erzeugt wurden. **(6P.)**

3. Das Unterprogramm **inserare** hat zwei Parameter:

- **n**, durch den es eine natürliche Zahl bekommt ( $2 \leq n \leq 20$ );
- **a**, durch den es ein eindimensionales Feld bekommt, das eine Folge von **n** natürlichen Zahlen speichert, jede mit höchstens 4 Ziffern. Wenigstens ein Element des Feldes ist Paarzahl.

Das Unterprogramm verändert das Feld, so dass nach jedem geraden Bestandteil der Folge der Wert **2011** eingefügt wird und durch alle Parameter **n** und **a**, die aktualisierten Werte der erhaltenen Daten liefert.

Schreibt in **Pascal** Sprache die vollständige Definition des Unterprogramms und die nötigen Datentypen.

**Beispiel:** wenn **n=7** und **a=(1, 4, 5, 3, 82, 6, 2)** dann ist nach dem Aufruf, **n=11** und **a=(1, 4, 2011, 5, 3, 82, 2011, 6, 2011, 2, 2011)**.

**(10P.)**

4. Es werden von der Tastatur zwei natürliche Zahlen **s1** und **s2** ( $0 < s1 \leq 18$ ,  $0 \leq s2 \leq 18$ ) eingelesen. Es wird verlangt, dass man in der Datei **BAC.TXT** auf je einer Reihe, in streng steigender Reihenfolge, alle natürlichen Zahlen, mit genau 5 Ziffern schreibt, für welche die Summe der ersten zwei Ziffern gleich mit **s1** ist und die Summe der letzten zwei Ziffern gleich mit **s2** ist. Um die angegebenen Zahlen zu bestimmen wird ein, im Bezug auf die Laufzeit, effizienter Algorithmus benützt.

**Beispiel:** wenn **s1=8**, und **s2=7**, dann ist **35725** eine der Zahlen die die verlangte Eigenschaft berücksichtigen ( $3+5=8$  und  $2+5=7$ ).

- a) Beschreibt in der Umgangssprache den benötigten Algorithmus und erklärt worin seine Effizienz besteht. **(4P.)**  
b) Schreibt das dem beschriebenen Algorithmus entsprechende **Pascal** Programm. **(6P.)**