

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

Varianta 1

Filiera teoretică, profilul real, specializarea științe ale naturii

- 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. $x==1 \ \&\& \ x==2 \ \&\& \ x==3$

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

c. $x==1 \ || \ x==2 \ || \ x==3$

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.)**

(30 Punkte)

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

1. Bestimmt einen C/C++ Ausdruck der den Wert 1 hat für jedwelche streng positive Zahl die in der reellen Variable x gespeichert ist. (4P.)
- a. $x - \text{floor}(x) != 0$ b. $x + \text{floor}(x) == 0$
- c. $x - \text{floor}(x) >= 0$ d. $x + \text{floor}(x) <= 0$
2. Sei die untere Sequenz, in der alle Variablen vom Typ ganz sind und von Null verschiedene natürliche Zahlen speichern.
- ```
y=1;
while ((y+1)*(y+1)<=x) y=y+1;
cout<<"....."; | printf("%d",);
```
- Um, nach dem Durchführen der Sequenz, auf dem Bildschirm die größte, quadratische, natürliche Zahl aus dem Intervall  $[1,x]$ , anzuzeigen, ersetzt man die Auslassungspunkte mit:
- (4P.)
- a.  $y$                       b.  $y*y$                       c.  $(y-1)*(y-1)$                       d.  $(y+1)*(y+1)$

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

3. Die reellen Variablen  $\mathbf{x_A}$  und  $\mathbf{y_A}$  speichern die Abszisse, beziehungsweise die Ordinate eines Punktes im  $\mathbf{xOy}$  Koordinatensystem, und die reellen Variablen  $\mathbf{x_B}$  und  $\mathbf{y_B}$  speichern die Abszisse, beziehungsweise die Ordinate eines anderen Punktes in demselben Koordinatensystem.  
Schreibt eine C/C++-Anweisungssequenz die am Bildschirm die Quadratlänge der Strecke mit Extremitäten in den obigen Punkten anzeigt. **(6P.)**
4. Es wird eine natürliche Zahl  $\mathbf{n}$  gelesen. Schreibt den kleinsten natürlichen Wert  $\mathbf{x}$  ( $\mathbf{x \geq n}$ ) der als Produkt von zwei aufeinanderfolgenden natürlichen Zahlen erhaltbar ist.  
**Beispiel:** wenn  $\mathbf{n=10}$  dann  $\mathbf{x=12}$  ( $\mathbf{12=3*4}$ ).  
**a)** Schreibt in Pseudocode den Algorithmus, welcher die gegebene Aufgabe löst. **(10P.)**  
**b)** Beschreibt die Rolle aller Variablen die im Algorithmus vom Punkt **a)** vorkommen und gebt die Eingabe -, beziehungsweise Ausgabedaten der gegebenen Aufgabe an. **(6P.)**

**THEMA III**

**(30 Puncte)**

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

1. In der unteren Anweisungssequenz sind alle Variablen vom Typ ganz.

```
for(i=1;i<=5;i++)
{ for(j=1;j<=5;j++)
 cout<<.....<<' '; | printf("%d ",.....);
 cout<<endl; | printf("\n");
}
```

|   |   |   |   |   |
|---|---|---|---|---|
| 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 |

Bestimmt einen Ausdruck der die Auslassungspunkte ersetzen kann, so dass nach dem Durchführen der erhaltenen Sequenz, die Werte aus der oberen Abbildung, in dieser Reihenfolge, angezeigt werden. **(4P.)**

- a.  $i+j\%5$       b.  $i\%5+j$       c.  $(i+j)\%5$       d.  $i\%5+j\%5$

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

2. Seien die eindimensionalen Felder  $a=(100,89,9,5,3)$  und  $b=(89,10,9,8,2,1)$ . Man erhält das Feld  $c$  durch die fallende Mischsortierung der Elementen der Felder  $a$  und  $b$ . Schreibt die Elemente des Feldes  $c$ , in der Reihenfolge in der sie erscheinen. **(6P.)**
3. Schreibt ein C/C++ Programm das von Tastatur eine natürliche Zahl  $n$  ( $2 < n < 50$ ), und die  $n$  Elementen eines eindimensionalen Feldes einliest. Die Elemente des Feldes sind höchstens vierstellige ganze Zahlen. Wenigstens ein Element des Feldes ist streng positiv. Das Programm verändert das Feld, so dass nach jedem streng positiven Element, ein Element mit demselben Wert hinzugefügt wird. Das Programm zeigt auf dem Bildschirm den aktualisierten Wert vom  $n$ , und danach, in einer neuen Zeile, die Elemente des veränderten Feldes, getrennt durch je ein Leerzeichen, an.  
**Beispiel:** für  $n=6$  und das Feld  $(4, -5, 0, 9, 9, -2)$ ,  
werden  $n=9$  und das Feld  $(4, 4, -5, 0, 9, 9, 9, 9, -2)$  angezeigt. **(10P.)**
4. Eine wenigstens zweistellige natürliche Zahl heißt **2-ordonat** wenn alle ihre Ziffern in steigender Reihenfolge sind und der Absolutbetrag der Differenz jedwelter ihrer zwei aufeinanderfolgenden Ziffern gleich mit 2 ist.  
**Beispiel :** die Zahl 2468 ist 2-ordonat, aber die Zahl 131 ist nicht 2-ordonat.  
Schreibt in die Datei **BAC.TXT** alle natürliche Zahlen die **2-ordonate** sind. Jede Zahl wird auf je eine Zeile der Datei geschrieben.  
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 begründet seine Effizienz. **(4P.)**  
b) Schreibt das dem beschriebenen Algorithmus entsprechende C/C++-Programm. **(6P.)**