

Examenul național de bacalaureat 2022

Proba E. d)
Informatică
Limbajul C/C++

Varianta 1

Filieră teoretică, profil real, specializare științe ale naturii

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei ore.
- Identificatorii utilizați în rezolvări 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). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

THEMA I (20 Puncte)

Für Punkt 1 bis 5 schreibt auf das Prüfungsblatt den Buchstaben welcher der richtigen Antwort entspricht. Jede richtige Antwort wird mit 4 Punkten bewertet.

1. Gebt den Wert des nebenstehenden C/C++ Ausdrucks an. | 22/20*20/2
- a. 0 b. 0.55 c. 10 d. 55
2. In der nebestehenden Sequenz sind alle Variablen vom Typ ganz und die 20 eingelesenen Zahlen sind natürlich. Gebt die Anweisungen an, die die Auslassungszeichen ersetzen können, sodass nach dem Durchlauf der erhaltenen Sequenz die Variable **strM** den Wert 1 hat, wenn alle eingelesenen Werte streng kleiner als 2022 waren oder 0 sonst.
- ```
strM=1;
for(i=1;i<=20;i++)
{ cin>>k; | scanf("%d",&k);

}
```
- a. **if(k<2022) strM=1;**  
  **else strM=0;**
- b. **if(k>=2022) strM=0;**  
  **else strM=1;**
- c. **if(k<2022) strM=1;**
- d. **if(k>=2022) strM=0;**
3. Um zu überprüfen, ob in einem eindimensionalen Feld sich das Element mit dem Wert **x=18** befindet, wird die Binärsuchmethode angewendet und die Reihenfolge der Elemente des Feldes, deren Werte mit dem Wert von **x**, während der Verwendung der gegebenen Methode verglichen werden, ist: **21, 16, 18**. Gebt die Reihenfolge der Werte, die Elemente des Feldes sein können, in der Reihenfolge in der sie im Feld erscheinen, an.
- a. (16, 17, 18, 19, 21, 50, 65)
- b. (4, 16, 18, 21, 52, 63, 70)
- c. (7, 9, 10, 16, 18, 19, 21)
- d. (16, 17, 18, 21, 30, 49, 50)
4. Die Variable **x** speichert eine, von Null verschiedene, natürliche Zahl. Gebt den maximalen Wert des nebenstehenden C/C++ Ausdrucks an. | **ceil(1+sqrt(x%100))**
- a. 10                                      b. 11                                      c. 99                                      d. 100
5. In der nebenstehenden C/C++ Sequenz sind alle Variablen vom Typ ganz. Gebt den Ausdruck an, der die Auslassungspunkte ersetzen kann, sodass nach dem Durchlaufen der erhaltenen Sequenz, die Variable **r** den Wert 20, die den Quotient der ganzen Teilung von 2022 auf 100 darstellt, speichert. (4p.)
- ```
r=0; x=100; y=2022;
do
{ y=y-x;
  r=.....;
}while(x<=y);
```
- a. **r+1** b. **r-1** c. **r/2** d. **r*2**

THEMA II (40 Puncte)

1. Der nebenstehende Algorithmus ist im Pseudocode dargestellt. Man beschriftet mit **a%b** den Rest der Teilung der natürlichen Zahl **a** durch die natürliche, von Null verschiedenen, Zahl **b** und mit **[c]** den ganzen Teil der reellen Zahl **c**.
- a. Schreibt was nach dem Durchlaufen des Algorithmus angezeigt wird, wenn die Zahl **56** eingelesen wird. (6P.)
- b. Schreibt zwei Zahlen aus dem Intervall **[10, 99]**, die so eingelesen werden können, sodass für jede dieser, nach dem Durchlaufen des Algorithmus, der Wert **1** angezeigt wird. (6P.)
- ```
lese n
(natürliche Zahl)
i←2; k←0
solange n≥i wiederhole
{ solange n%i=0 wiederhole
 { k←k+1
 n←[n/i]
 }
 wenn i=2 dann i←i+1
 sonst i←i+2
}
```

- c. Schreibe das dem gegebenen Algorithmus entsprechende C/C++ Programm. (10P.) ■ schreibe k
- d. Schreibe in Pseudocode einen, mit dem gegebenen, äquivalenten Algorithmus, in dem ihr die zweite **solange...wiederhole** Struktur entsprechend mit einer fußgesteuerten Wiederholung ersetzt. (6P.)
2. Die eindimensionalen Felder **A** und **B** haben die Werte **A=(21, 18, 16, 7, 5)** und **B=(49, 15, 14, 3, 2)** und sind in steigender Reihenfolge mischsortiert, durchlaufend von **rechts nach links**. Schreibe alle Werte des Feldes **B**, die mit dem Element mit dem Wert **16** des Feldes **A** verglichen werden, während der verwendeten Methode, an. (6P.)
3. Für ein Material werden spezifische Daten gespeichert: Name (ein Großbuchstabe des englischen Alphabets), Einheitspreis und Menge (natürliche Zahlen). Die Variablen **denumire1**, **pret1** und **cantitate1** speichern spezifische Daten für ein Material und die Variablen **denumire2**, **pret2** und **cantitate2** speichern spezifische Daten für ein anderes Material. Deklariere die Variablen **denumire1** und **denumire2** und schreibe eine C/C++ Anweisungssequenz, sodass nach dem Durchlaufen dieser auf dem Bildschirm der entsprechende Name des Materials, der den größeren Gesamtkosten (das Produkt zwischen Einheitspreis und Menge) hat, oder die Nachricht **egal**, wenn die zwei Materiale gleiche Gesamtkosten haben. (6P.)

### THEMA III

(30 Punkte)

1. Es wird eine natürliche Zahl, **n** ( $n \in [10, 10^9]$ ) eingelesen, in der es keine Sequenzen von höchstens zwei gleichen Ziffern, die sich auf aufeinanderfolgenden Positionen befinden, gibt und es wird verlangt, es soll die aus **n** erhaltene Zahl durch die Ersetzung jeder **22** Sequenz mit je einer **20** Sequenz angeschrieben werden. Wenn keine einzige Sequenz ersetzt wird, wird die unveränderte Zahl angeschrieben. Schreibe in Pseudocode den Lösungsalgorithmus für die erläuterte Aufgabe. (10P.)  
**Beispiel:** wenn **n=202233228** wird **202033208** angeschrieben.
2. Um das Phänomen der Erosion am Schwarzen Meer zu verhindern haben die Autoritäten entschlossen einen Strand zu versanden. Der Strand wurde in quadratförmige Parzellen mit der Seite von **1** Meter eingeteilt, die als Elemente eines eindimensionalen Feldes verteilt sind und das Niveau jeder Parzelle, gemessen in Meter von dem Meeresniveau, speichern. Die Versandung wird nur auf den Parzellen durchgeführt, die das Niveau streng kleiner als das der niedrigsten benachbarten Parzelle haben, indem sie Sand hinzufügen, sodass die beiden dasselbe Niveau haben. Zwei Parzellen sind benachbart, wenn sie eine gemeinsame Seite haben. Schreibe ein C/C++ Programm, das von der Tastatur zwei natürliche Zahlen aus dem Intervall  $[2, 10^2]$ , **n** einliest, nachher die **n** Elemente eines eindimensionalen Feldes, natürliche Zahlen aus dem Intervall  $[1, 10]$ , die die Anzahl der Niveaus der Parzellen darstellen, in der Reihenfolge in der sie auf dem Strand verteilt sind. Das Programm zeigt auf dem Bildschirm die insgesamt hinzugefügte Sandmenge, gemessen in Kubikmeter, an. (10P.)  
**Beispiel:** für **n=12** und dem nebenstehenden Feld werden 

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 9 | 3 | 6 | 3 | 2 | 4 | 4 | 5 | 3 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|

 die markierten Parzellen versandet, um auf das entsprechende Niveau der Nachbarparzellen gebracht zu werden und es wird der Wert **7** ( $7=1+3+1+2$ ) angezeigt.
3. Die Datei **bac.txt** enthält natürliche Zahlen aus dem Intervall  $[1, 10^9]$ , folgendermaßen: auf der ersten Reihe die Zahl **x**, und auf der zweiten Reihe eine Folge von höchstens  $10^6$  Zahlen, steigend angeordnet. Die Zahlen, die sich auf derselben Reihe der Datei befinden sind durch je ein Leerzeichen getrennt. Schreibe auf dem Bildschirm die Anzahl der unterschiedlichen Werte aus der Folge an, die sich auf der zweiten Reihe der Datei im Intervall  $[1, x]$  befinden. Erstelle einen in Bezug auf den benutzten Speicherplatz und auf die Laufzeit effizienten Algorithmus. (2P.)  
**Beispiel:** wenn die Datei 

|                                               |
|-----------------------------------------------|
| 9                                             |
| 1 1 1 2 2 3 5 5 5 5 6 6 7 8 10 10 12 15 21 21 |

 nebenstehenden Inhalt hat, wird auf dem Bildschirm **7** angezeigt. (8P.)
- a. Beschreibe in Umgangssprache den erstellten Algorithmus und begründe seine Effizienz. (2P.)
- b. Schreibe das dem erstellten Algorithmus entsprechende C/C++ Programm. (8P.)