

THEMA II

(30 Punkte)

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

1. Sei ein ungerichteter Graph mit 6 Knoten und 9 Kanten. Die Anzahl der Kanten die eingefügt werden müssen, so dass der so erhaltene Graph vollständig ist, ist: **(4P.)**
a. 5 b. 6 c. 12 d. 15
2. Sei nebenstehende Anweisungssequenz in welcher die Variablen `i` und `j` vom Typ `ganz` sind, die Variable `a` vom Typ `char` ist und die Variable `s` kann eine Folge mit höchstens 20 Zeichen speichern. Die Folge, die nach dem Durchführen der Sequenz angeschrieben wird, ist: **(4P.)**
- ```
strcpy(s, "bacalaureat");
j=strlen(s);
for(i=0; i<3; i++)
 if(s[i]!=s[j-i-1])
 { a=s[i];
 s[i]=s[j-i-1];
 s[j-i-1]=a;
 }
cout<<s; | printf("%s", s);
```
- a. bacalaureab      b. tacalaureab      c. tacrlauaeab      d. taalaurcab

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

3. Sei ein Baum mit Wurzel und 8 von 1 bis 8 nummerierten Knoten und den Kanten [1, 2], [1, 3], [2, 4], [3, 5], [3, 6], [6, 7], [6, 8]. Wenn der mit 1 nummerierte Knoten die Wurzel des Baumes ist und sich auf dem Niveau 0 des Baumes befindet, dann schreibt alle Knoten auf, die sich am Niveau 2 des gegebenen Baumes befinden. **(6P.)**
4. Die nebenstehend deklarierten Variablen `e1` und `e2`, speichern den Namen und das Geburtsdatum für je einen Schüler. Wenn man weiß, dass die Schüler unterschiedliche Geburtsjahre haben, dann schreibt die Anweisungssequenz durch die der Name des älteren Schülers am Bildschirm angezeigt wird. **(6P.)**
- ```
struct data_n {  
    int an, luna, zi;  
};  
struct elev{  
    char nume[20];  
    data_n d;  
} e1, e2;
```
5. Schreibt ein C/C++ Programm, das von der Tastatur eine natürliche Zahl `n` ($2 \leq n \leq 20$) einliest und ein bidimensionales Feld mit `n` Zeilen und `n` Spalten baut und speichert, das folgende Eigenschaften hat:
- Alle Elemente der Hauptdiagonale haben den Wert 0;
 - Jede Zeile enthält beginnend von der Hauptdiagonale von rechts nach links eine streng steigende Folge von aufeinanderfolgenden Zahlen und beginnend von der Hauptdiagonale von links nach rechts auch eine streng steigende Folge von aufeinanderfolgenden Zahlen.
- Das Programm zeigt am Bildschirm das erhaltene Feld an, jede Zeile des gebauten Feldes in je einer Bildschirmzeile, wobei die Elemente derselben Zeile durch je ein Leerzeichen getrennt sind.
- Beispiel:** wenn `n=5` ist, dann wird am Bildschirm das nebenstehende Feld angezeigt. **(10P.)**
- | | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 |
| 1 | 0 | 1 | 2 | 3 | |
| 2 | 1 | 0 | 1 | 2 | |
| 3 | 2 | 1 | 0 | 1 | |
| 4 | 3 | 2 | 1 | 0 | |

THEMA III

(30 Punkte)

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

1. Die Backtracking-Methode verwendend erhält man alle geraden dreistelligen Zahlen deren Ziffern zur Menge {7, 8, 1, 6, 2, 3} gehören. Die ersten 4 erzeugten Lösungen sind in dieser Reihenfolge: 778, 776, 772, 788. Die 8. erzeugte Lösung ist: **(4P.)**
- a. 712 b. 716 c. 718 d. 782

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

2. Sei das nebenstehend definierte Unterprogramm **f**. Schreibt zwei natürliche Zahlen aus dem offenen Intervall (20,30), die von den ganzen Variablen **x1** beziehungsweise **x2** gespeichert werden können, so dass der Wert von **f(x1,3)** gleich 29 und der Wert von **f(3,x2)** gleich 1 ist. **(6P.)**
- ```
int f(int a, int b)
{
 if(a>=b)
 return (a-b)+f(a/b,b);
 return 1;
}
```
3. Sei die Folge **s**, die nach nebenstehender Regel gebaut ist.
- $$s_n = \begin{cases} 1 & \text{dacă } n \leq 2 \\ 3 \cdot s_{n-1} - s_{n-2} & \text{dacă } n > 3 \end{cases}$$
- Das Unterprogramm **sir** bekommt durch den Parameter **n** eine natürliche Zahl ( $3 < n < 20$ ) und liefert durch den Parameter **a** ein eindimensionales Feld, das die ersten **n** Glieder der Folge **s** speichert, die nach der obigen Regel definiert ist, so dass die ungeraden Zahlen die ersten Positionen im Feld besetzen und die geraden Zahlen werden nach den ungeraden gespeichert.
- Schreibt die vollständige Definition des Unterprogramms **sir**.
- Beispiel:** wenn **n=6** ist, dann sind die ersten sechs Glieder der Folge: 1, 1, 2, 5, 13, 34, und nach dem Aufruf kann das gebaute Feld (1,1,5,13,2,34) sein. **(10P.)**
4. Die Datei **bac.in** enthält in der ersten Zeile eine natürliche Zahl **n** ( $3 < n < 1000$ ) und in der nächsten Zeile eine Folge von **n** natürlichen, unterschiedlichen höchstens neunstelligen Zahlen. Die Zahlen der Folge sind durch je ein Leerzeichen getrennt und wenigstens drei von ihnen enden mit der Ziffer 5.
- a) Schreibt ein C/C++ Programm, das alle Zahlen der Datei liest und einen effizienten Algorithmus in Bezug auf die Laufzeit und den benötigten Speicherplatz verwendet und die drei größten Zahlen der Folge bestimmt und am Bildschirm anschreibt, deren letzte Ziffer 5 ist. Die bestimmten Zahlen werden in steigender Reihenfolge und durch je ein Leerzeichen getrennt angeschrieben. **(6P.)**
- Beispiel:** wenn die Datei **bac.in** nebenstehenden Inhalt hat, dann werden am Bildschirm in dieser Reihenfolge die Zahlen: 15 25 85 erscheinen.
- ```
10
97 5 11 1 8 6 85 3 25 15
```
- b) Beschreibt kurz in der Umgangssprache (3-4 Zeilen) den bei Punkt a) verwendeten Algorithmus und erklärt worin seine Effizienz besteht. **(4P.)**