

THEMA II

(30 Puncte)

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

1. Die Variable x speichert eine reelle Zahl aus dem Intervall $[1,2]$. Gebt die mögliche Anzahl der Werte des nebenstehenden C/C++ Ausdruckes an.

`ceil(x)+floor(x)`

(4P.)

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

2. In den unterstehenden, mit $s1$, $s2$ und $s3$ notierten Sequenzen, sind alle Variablen vom Typ ganz und n und k speichern je eine von Null verschiedene natürliche Zahl.

```
//s1
nr=0;
for(i=n;i>0;i--)
{ p=i;
  while(p>=k) p=p-k;
  if(p==0) nr=nr+1;
}
```

```
//s2
nr=0;
for(i=1;i<=n;i++)
  if(i%k==0) nr=nr+1;
```

```
//s3
nr=n/k;
```

Die Variable nr speichert die Anzahl der Vielfachen von k aus dem Intervall $[1,n]$ nach dem unabhängigen Durchlaufen der Sequenzen:

(4P.)

- a. $s1$ und $s2$ b. $s1$ und $s3$ c. $s2$ und $s3$ d. $s1$, $s2$ und $s3$

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

3. Für ein Medikament wird der Preis, sowie auch die Daten über die aktive spezifische Substanz, dessen Dosis und Code gespeichert.

Die ganzen Variablen $pret1$, $doza1$ und $cod1$ speichern den Preis, beziehungsweise die Daten über die aktive spezifische Substanz eines Medikamentes und $pret2$, $doza2$ und $cod2$ speichern den Preis, beziehungsweise die Daten über die aktive spezifische Substanz eines anderen Medikamentes. Zwei unterschiedliche Medikamente können dieselbe aktive spezifische Substanz haben, aber nur in verschiedener Dosis.

Schreibt eine C/C++ Anweisungssequenz, so dass nach deren Durchlaufen auf dem Bildschirm der Preis des Medikamentes mit größerer Dosis an aktiver spezifischen Substanz angezeigt wird, wenn sie dieselbe Substanz enthalten, oder die Nachricht **ALTA CATEGORIE** im Gegenfall.

(6P.)

4. Es werden vier natürliche, von Null verschiedene Zahlen, n , x , y und r ($r < x < y < n$) eingelesen und es soll die Anzahl der natürlichen Werte aus dem Intervall $[1,n]$, für die sowohl der Rest der Teilung durch x , als auch der Rest der Teilung durch y gleich mit r sind, angeschrieben werden.

Beispiel: für $n=200$, $x=5$, $y=14$ und $r=2$, wird die Zahl 3 angezeigt (für die Zahlen 2, 72 und 142 ist sowohl der Rest der Teilung auf 5, als auch der Rest der Teilung auf 14 der Wert 2).

a) Schreibt in Pseudocode den Lösungsalgorithmus für die erläuterte Aufgabe. **(10P.)**

b) Bestimmt die Rolle aller Variablen, welche im, bei Punkt a), dargestellten Algorithmus auftreten und gebt die Eingabedaten, beziehungsweise die Ausgabedaten der erläuterten Aufgabe an. **(6P.)**

TEMA III

(30 Puncte)

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

1. Die Variablen i und j sind vom Typ ganz. Schreibt einen Ausdruck der die Auslassungspunkte ersetzen kann, so dass nach dem Durchlaufen der erhaltenen Sequenz auf dem Bildschirm die nebenstehenden Zahlen angezeigt werden.

```
for(i=0;i<9;i++)
{ for(j=0;j<9;j++)
    if(i==j) cout<<"0 "; | printf("0 ");
    else cout<<.....<<' '; | printf("%d ",...);
    cout<<endl; | printf("\n");
}
```

(4P.)

0	1	2	3	4	5	6	7	0
1	0	3	4	5	6	7	0	1
2	3	0	5	6	7	0	1	2
3	4	5	0	7	0	1	2	3
4	5	6	7	0	1	2	3	4
5	6	7	0	1	0	3	4	5
6	7	0	1	2	3	0	5	6
7	0	1	2	3	4	5	0	7
0	1	2	3	4	5	6	7	0

- a. $(i-j)\%8$ b. $(i+j)\%8$ c. $(i-j)/8$ d. $(i+j)/8$

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

2. In der nebenstehenden Sequenz sind alle Variablen vom Typ ganz und die eingelesenen Zahlen sind natürlich. Schreibt einen Ausdruck der die Auslassungspunkte ersetzen kann, so dass nach dem Durchlaufen der erhaltenen Sequenz, die Variable `bun` den Wert 0 haben soll, wenn alle eingelesenen Werte verschieden von 2018 sind oder den Wert 1 sonst.

(6P.)

```
bun=.....;
for(i=1;i<=20;i++)
{
    cin>>k; | scanf("%d",&k);
    .....
}
```

3. Es seien zwei Folgen von natürlichen Werten, s und c mit gleicher Anzahl von Gliedern. Man nennt **Kodifizierung** von s anhand des Code c die Operation die eine neue Folge baut, in der Anfangs der erste Wert aus s kopiert wird, nachher die restliche Folge s , von links nach rechts durchgehend, wird der kurrente Wert aus s am Anfang der neuen Folge eingefügt, wenn das Element von derselben Position in c gerade ist oder am Ende der neuen Folge im Gegenfall.

Beispiel: wenn s die Folge 7, 3, 6, 2, 8 und c der Code 0, 12, 5, 23, 14 ist, erhält man die kodifizierte Folge 8, 3, 7, 6, 2 (anfangs enthält die Folge das Glied 7, nachher werden die Glieder 3 und 8 am Anfang, in der Reihenfolge in der man s durchgeht, und am Ende der Rest der Glieder).

Schreibt ein C/C++ Programm, das von der Tastatur natürliche Zahlen aus dem Intervall $[0, 10^2]$ einliest: ns, nc ($ns \geq 2, nc \geq 2$) die ns Glieder der Folge s , danach die nc Glieder der Folge c . Das Programm zeigt auf dem Bildschirm, durch je ein Leerzeichen getrennt, die Glieder der neuen Folge, erhalten durch die Kodifizierung von s anhand von c , wenn die beiden Folgen dieselbe Anzahl von Gliedern haben, sonst die Nachricht `cod incorrect`.

Beispiel: wenn $ns=5, nc=5, s=(7, 3, 6, 2, 8)$, und $c=(0, 12, 5, 23, 14)$, wird:

8 3 7 6 2 angezeigt.

(10P.)

4. Man nennt **ungleiche Sequenz** einer Folge von natürlichen Zahlen, eine Unterfolge dieser, gebildet aus den Gliedern, die sich auf aufeinanderfolgenden Positionen, in der gegebenen Folge befinden, mit der Eigenschaft, dass jedwelche zwei Nachbarglieder verschieden sind. Die Länge der Sequenz ist gleich mit der Anzahl ihrer Glieder.

Die Datei `bac.txt` enthält eine Folge von höchstens 10^6 natürlichen Zahlen aus dem Intervall $[0, 9]$. Die Zahlen sind durch je ein Leerzeichen getrennt und in der Folge sind wenigstens zwei verschiedene Glieder auf aufeinanderfolgenden Positionen.

Man soll auf dem Bildschirm die maximale Länge der ungleichen Sequenz der Folge, die sich in der Datei befindet, anschreiben. Erzeugt einen in Bezug auf den benötigten Speicher und auf die Laufzeit effizienten Algorithmus.

Exemplu: wenn die Datei `bac.txt` die Zahlen

7 7 1 3 1 1 7 3 3 3 7 3 9

Enthält, dann wird auf dem Bildschirm der Wert 4 angeschrieben.

- a) Beschreibt in Umgangssprache den entworfenen Algorithmus und begründet seine Effizienz. (2P.)

- b) Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. (8P.)