

Klausur zum Eignungsfeststellungsverfahren
für das
Informatikstudium
an der
Ludwig-Maximilians-Universität München

12. August 2002, 13-18 Uhr

Name:

Vorname:.....

Geburtsdatum:

Punktzahlen (bitte frei lassen):

1	2	3	4	5	6	7	8	9	10	
× 1	× 1	× 2	× 1	× 2	× 2	× 4	× 4	× 4	× 0,5	Σ

Die folgenden Aufgaben von unterschiedlichem Schwierigkeitsgrad werden alle durch Ankreuzen beantwortet. Jede richtig angekreuzte Antwort ergibt einen Punkt; für jede falsch angekreuzte Antwort wird ein Punkt abgezogen. Es können mehrere Antworten richtig sein; die einzelnen Aufgaben haben nach ihrem Schwierigkeitsgrad unterschiedliches Gewicht.

Lesen Sie insbesondere die langen Aufgaben genau und in Ruhe durch, bevor Sie sich zu einer Antwort entschließen.

Viel Erfolg!

1 Hotelkosten

In einem Hotel kostet die Übernachtung 70 €. Hinzu kommen möglicherweise sonstige Kosten (Restaurant, Telefon, etc.) Auf all das werden noch 16% MWSt (= VAT, sales tax) erhoben. Gesucht ist eine Formel für den Rechnungsbetrag R (in €) als Funktion der Zahl der Nächte N und den sonstigen Kosten S (in €). Kreuzen Sie die richtige Formel an:

- $R = 0,16 \cdot (70N + S)$
- $R = 16 \cdot (70N + S)$
- $R = 1,16 \cdot (70N + S)$
- $R = 70N + S + 0,16$
- $R = 70N + S + 16\%$
- $R = 70N^S - 16\pi$

2 Magisches Quadrat

Ein magisches Quadrat ist eine 3×3 Tabelle mit 9 Einträgen, die die Ziffern 1 bis 9 enthalten, so dass die Summe der Einträge in jeder Zeile, jeder Spalte und jeder Diagonalen jeweils 15 ergibt. Die folgende Tabelle ist *kein* magisches Quadrat, kann aber durch Vertauschen zweier Einträge in ein solches überführt werden. Markieren Sie die zu vertauschenden Einträge mit jeweils einem Kreuz.

8	1	6
5	3	7
4	9	2

3 Würfel

Wie lang ist die Raumdiagonale in einem Würfel der Kantenlänge 7? Kreuzen Sie die richtige Antwort an.

- $7\sqrt[3]{2}$
- $2\sqrt{7}$
- $7\sqrt{3}$
- $7\sqrt{5}$
- $7\sqrt{2}$
- 1 Lichtjahr

4 Herzen

Für beliebige reelle Zahlen x, y definieren wir $x \heartsuit y = x + y^2$, also zum Beispiel $5 \heartsuit 4 = 21$. Kreuzen Sie die *wahren* Aussagen über die Verknüpfung \heartsuit an:

- $6 \heartsuit 2 = 10$
- $6 \heartsuit 2 = 9$
- $a \heartsuit 1 \geq 1 \heartsuit a$ für alle $a \in \mathbb{R}$
- $a \heartsuit 1 < 1 \heartsuit a$ für alle $a \in \mathbb{R}$
- Es gibt ein $a \in \mathbb{R}$ so dass $a \heartsuit 1 < 1 \heartsuit a$.
- Die Anzahl der Paare (x, y) mit $x \heartsuit y = 10$, wobei $x, y \in \mathbb{N}_0$, ist 4.
- Die Anzahl der Paare (x, y) mit $x \heartsuit y = 10$, wobei $x, y \in \mathbb{N}_0$, ist 2.
- Die Anzahl der Paare (x, y) mit $x \heartsuit y = 10$, wobei $x, y \in \mathbb{N}_0$, ist 5.

Bemerkung: $\mathbb{N}_0 = \{0, 1, 2, 3, 4, \dots\}$.

5 Fakultät

Das Produkt der ersten n natürlichen Zahlen wird mit $n!$ bezeichnet, also etwa $6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$. Was ist $\frac{1001!}{999!}$? Kreuzen Sie die richtige Antwort an.

- 1001000
- 1001001
- 1001
- 19518710109889195794001900000
- 1,002002

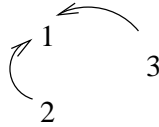
6 Quadrate

Bestimmen Sie $1.000.000.000.000.001^2 - 999.999.999.999.999^2$. Kreuzen Sie die richtige Antwort an.

- 1.000.000.000.000.000
- 2.000.000.000.000.000
- 4.000.000.000.000.000
- 1.000.000.000.000
- 2.000.000.000.000
- 4.000.000.000.000
- 4

7 Graphenfärbung

Dies ist ein Graph:



Er besteht aus den Knoten 1, 2 und 3, sowie den Kanten $(2, 1)$ und $(3, 1)$. Bitte verwechseln Sie das nicht mit den Ihnen bekannten *Funktionsgraphen*, z.B. Parabeln oder Sinuskurven.

Bei einer *Färbung* eines Graphen ordnet man den Knoten Buchstaben („Farben“) zu, z.B. färben wir den obigen Graphen mit den zwei Farben r und b : $1 \mapsto r$, $2 \mapsto b$, $3 \mapsto b$.

Dabei muss stets die *Färberegeln* beachtet werden: wenn zwei Knoten durch eine Kante verbunden sind, dürfen sie nicht dieselbe Farbe erhalten.

Wieviele Farben benötigt man mindestens zur Färbung eines Graphen mit den Knoten $1, 2, 3, \dots, n$ und den Kanten: $(1, 2), (2, 3), \dots, (n-1, n), (n, 1)$ wenn n ungerade ist?

- 2 Farben
- 3 Farben
- 4 Farben
- n Farben
- $n - 1$ Farben
- 256 Farben

8 Geheimschrift

Wir betrachten folgendes Verschlüsselungsverfahren (eine vereinfachte Version der „Playfair Cipher“ von Ch. Wheatstone):

Zeichen sind die Buchstaben A-Z (keine Kleinbuchstaben, Umlaute, etc.).

Gegeben ist ein Schlüsselwort.

In eine 5×5 Tabelle wird nun zeilenweise, beginnend links oben, das Schlüsselwort eingetragen, gefolgt von den Buchstaben des Alphabets, welche nicht im Schlüsselwort vorkommen. Der Buchstabe J wird nicht eingetragen, sodass die 25 Felder gerade ausreichen.

Um ein Wort zu verschlüsseln, werden die Buchstaben des Wortes in aufeinanderfolgenden Paaren verschlüsselt. Befinden sich beide Buchstaben in derselben Spalte, so werden sie durch die beiden jeweils darunterliegenden Buchstaben verschlüsselt. Befinden sich die beiden Buchstaben in derselben Zeile, so werden sie durch die jeweils

rechts danebenliegenden Buchstaben ersetzt. Andernfalls bestimmen die beiden Buchstaben schräg gegenüberliegende Ecken eines Rechtecks, die beiden anderen Ecken bilden dann die Verschlüsselung, wobei die Verschlüsselung jedes einzelnen Buchstaben in derselben Zeile wie dieser liegt.

Bitte beachten Sie, dass diese Vorschrift aus mehreren Gründen *unvollständig* ist. Das Schlüsselwort laute BART. Kreuzen Sie die wahren Aussagen an:

- Die Verschlüsselung von GEIFER lautet HFDLFA.
- Die Verschlüsselung von GEIFER lautet HFLDFA.
- Die Verschlüsselung von GEIFER lautet AFLDXZ.
- Die Verschlüsselung von GEIFER lautet HFLDFU.
- Die Entschlüsselung von UTDUHK lautet SCHOPF.
- Die Entschlüsselung von UTDUHK lautet BRAUEN.
- Die Entschlüsselung von UTDUHK lautet SCHOEN.
- Die Entschlüsselung von UTDUHK lautet SCHONE.
- Wörter, die ein J enthalten, lassen sich nicht verschlüsseln.
- Manche Wörter lassen sich auf zwei Arten verschlüsseln.
- Das Wort PUTZ lässt sich nicht verschlüsseln.

9 Programme

Eine Zahlenfolge G_n ist durch die Vorschrift

$$\begin{aligned} G_0 &= 1 \\ G_1 &= 1 \\ G_{n+2} &= G_n + 2G_{n+1} \end{aligned}$$

definiert, also z.B. $G_3 = 7$. Folgendes Programm einer hypothetischen Programmiersprache gestattet es, den Wert G_n zu berechnen:

```

Eingabe von n
u:=1
v:=1
Schleife wenn n>1
dann n:=n-1
w:=v
v:=u+2*v
u:=w
springe nach ``Schleife``
sonst wenn n=0
dann Ausgabe von u
sonst Ausgabe von v

```

Erklärung des Programms Hier sind u, v, w, m, n Programmvariablen, die während des Programmablaufs ihren Wert ändern nach Maßgabe der Wertzuweisungen wie $u:=0$. Die Programmzeilen werden der Reihe nach abgearbeitet mit folgenden Ausnahmen: beim Ausführen des `springe` nach Befehls wird zu der mit `Schleife` gekennzeichneten Programmzeile gesprungen, beim Ausführen der `wenn` Anweisung wird entweder zu der mit `dann` oder der mit `sonst` gekennzeichneten Programmzeile gesprungen, je nachdem ob die Bedingung zutrifft oder nicht. In der ersten `wenn` Anweisung lautet diese Bedingung $n>1$.

Ihre Aufgabe Die Zahlenfolge H_n ist durch die folgende Vorschrift definiert:

$$\begin{aligned} H_0 &= 1 \\ H_1 &= 2 \\ H_2 &= 3 \\ H_{n+3} &= H_n + H_{n+1} - H_{n+2} \end{aligned}$$

Was muss man anstelle von XXXX in folgendem Programm einsetzen, damit im gleichen Sinne H_n berechnet wird?

```

Eingabe von n
u:=1
v:=2
w:=3
Schleife wenn n>2
dann n:=n-1
XXXX
springe nach ``Schleife``
sonst wenn n=0
dann Ausgabe von u
sonst wenn n=1
dann Ausgabe von v
sonst Ausgabe von w

```

Kreuzen Sie in der folgenden Auswahl die richtige Einsetzung für XXXX an.

- $h:=w$
 $v:=u+v-w$
 $u:=w$
 $v:=h$
- $h:=w$
 $w:=u+v-w$
 $u:=v$
 $v:=h$
- $w:=u+v-w$
 $u:=v$
 $v:=w$

- h := w
 - w := v
 - u := u + v - w
 - v := h

- h := w
 - w := v
 - u := u + v + w + 3, 14159
 - v := w

10 Englisch

Lesen Sie folgenden Text aus der "Javacard Spezifikation" aufmerksam durch:

There are three main types of memory on a smart card, they are:

ROM (Read-Only Memory) - contains code and data that is read-only and cannot be modified. Information stored in ROM persists even after power to the card is disconnected.

RAM (Random Access Memory) - is fast, volatile memory. Any information in RAM is lost when power to the card is disconnected. A typical Java Card platform implementation uses RAM for the frame and operand stacks and for storing temporary data.

EEPROM (Electrically Erasable Programmable Read Only Memory) - is like ROM in that information in this type of memory persists across power sessions (other forms of non-volatile memory are also used in smart cards, for example, Flash and battery-backed RAM). EEPROM has the added advantage over ROM of being both readable and writeable (like RAM), although writing to EEPROM is slower than writing to RAM and EEPROM is subject to wear. After a large number of writes to a particular byte, typically more than 100,000, the data integrity of that byte in EEPROM may fail. See the chip manufacturer's specification for details.

The applet developer should ensure that temporary fields that are updated frequently are components of transient arrays. This reduces potential wear on persistent memory and guarantees better write performance. As a rule of thumb, if a temporary field is being updated multiple times for every Application Protocol Data Unit (APDU), the applet developer should move it into a transient array.

Kreuzen Sie die wahren Aussagen zum Text an:

- Die Speicherarten in einer Smartcard sind RAM, ROM, EEPROM.
- APDU ist eine der Speicherarten in einer Smartcard.
- Nach 200.000 Schreibzugriffen kann ein EEPROM byte fehlerhaft werden.
- Nach 200.000 Lesezugriffen kann ein EEPROM byte fehlerhaft werden.
- Nach 200.000 Schreibzugriffen muss ein EEPROM byte ersetzt werden.
- EEPROM bedeutet "elektrisch lesbarer programmierbarer Nur-lese-Speicher".

- EEPROM bedeutet “elektrisch lesbarer programmgesteuerter Nur-lese-Speicher”.
- EEPROM bedeutet “elektrisch löschbarer programmierbarer Nur-lese-Speicher”.
- Der RAM Speicher ist nichtflüchtig.
- Der ROM Speicher ist nichtflüchtig.
- Häufig geänderte temporäre Felder sollen in einem transienten Array abgelegt werden, um die Schreibleistung zu erhöhen.
- Häufig geänderte temporäre Felder sollen in einem transienten Array abgelegt werden, um Datenverlust bei Stromausfall zu vermeiden.
- Häufig geänderte temporäre Felder sollen in einem transienten Array abgelegt werden, um die Abnutzung des persistenten Speichers zu reduzieren.
- Häufig geänderte temporäre Felder sollen in einem transienten Array abgelegt werden, um den Potentialbeitrag des persistenten Speichers zu reduzieren.