

--	--	--	--	--	--	--	--

Bitte hier unbedingt Matrikelnummer und Adresse eintragen, sonst keine Bearbeitung möglich.

Postanschrift: FernUniversität - 58084 Hagen

---

(Name, Vorname)

---

(Straße, Nr.)

---

(PLZ, Wohnort)

---

## KLAUSUR zum Kurs Elementare Zahlentheorie mit Maple (01202) SoSe 2013

**DATUM:** 24.08.2013  
**UHRZEIT:** 10.00 - 12.00 Uhr  
**KLAUSURORT:**

### Bearbeitungshinweise

(Bitte vor Arbeitsbeginn durchlesen!)

1. Schreiben Sie Ihre Klausur bitte nicht mit Bleistift.
2. Füllen Sie bitte das Adressfeld leserlich und vollständig aus, und schreiben Sie Ihren Namen und Ihre Matrikelnummer auf jedes Lösungsblatt, das Sie abgeben.
3. Die Reihenfolge, in der Sie die Aufgaben/Teilaufgaben lösen, ist Ihnen freigestellt. Kreuzen Sie in der Tabelle (s.u.) an, welche Aufgaben Sie bearbeitet haben.
4. Bei jeder Aufgabe ist die erreichbare Höchstpunktzahl vermerkt. Sie haben die Klausur bestanden, wenn Sie **40** Punkte erreichen.
5. Als Hilfsmittel erlaubt ist ein **beidseitig handbeschriebenes DIN A4-Blatt**.
6. Weitere Hilfsmittel wie Bücher, Taschenrechner, Studienbriefe, weitergehende eigene Aufzeichnungen, (Tablet-)PCs, eBookreader etc. dürfen während der Klausur nicht benutzt werden. Ihre Benutzung sowie andere Täuschungsversuche führen dazu, dass Ihre Klausur mit 5 bewertet wird.

	<b>Bemerkungen:</b>

Aufgabe	1	2	3	4	5	6	7	8	Summe
<b>Bearbeitet</b>									
<b>max. Punktezahl</b>	10	10	11	10	10	9	8	12	80
<b>erreichte Punktezahl</b>									
<b>Korrektur</b>									

<b>Prüfergebnis/Note</b>	
--------------------------	--

## **HINWEIS:**

Erlaubtes Hilfsmittel ist nur

**ein beidseitig handbeschriebenes DIN A4-Blatt.**

Weitere Hilfsmittel wie z.B. Studienbriefe, Bücher, Taschenrechner, (Tablet-)PCs, eBookreader etc. sind nicht erlaubt.

**Klausur am 24.08.2013:****Aufgabenstellungen**

---

Die Lösungen aller Aufgaben müssen Sie begründen.

### **Aufgabe 1**

Beweisen Sie: Für jedes  $n \in \mathbb{N}$  mit  $\text{ggT}(n, 6) = 1$  ist  $n^2 - 1$  durch 24 teilbar.

[10 Punkte]

### **Aufgabe 2**

Sei  $p$  eine Primzahl. Wir betrachten die natürliche Zahl  $n = 2 \cdot 3 \cdot 5 \cdots p$ , also das Produkt aller Primzahlen bis einschließlich  $p$ . Zeigen Sie, dass die  $p - 1$  aufeinanderfolgenden natürlichen Zahlen

$$n + 2, n + 3, n + 4, \dots, n + p$$

zusammengesetzt sind.

[10 Punkte]

### **Aufgabe 3**

Bestimmen Sie durch Kongruenzrechnung den kleinsten positiven Rest, den man bei Division von  $2^{1000}$  durch 31 erhält.

Hinweis: Kleiner Satz von Fermat.

[11 Punkte]

### **Aufgabe 4**

1. Berechnen Sie  $\tau(60)$  und begründen Sie das Ergebnis.
2. Berechnen Sie  $\sigma(60)$  und begründen Sie das Ergebnis.
3. Unter welchen Bedingungen wird eine zahlentheoretische Funktion als multiplikativ bezeichnet?

[4 + 4 + 2 = 10 Punkte]

## Aufgabe 5

Beweisen Sie: Für jede ganze Zahl  $k$  existieren nur endlich viele pythagoreische Tripel, die  $k$  enthalten.

[10 Punkte]

## Aufgabe 6

Welche der drei Zahlen 27, 45 und 125 kann als Summe von zwei Quadraten dargestellt werden und warum bzw. warum nicht?

[9 Punkte]

## Aufgabe 7

Sie sehen nachfolgend eine Prozedur. Leider sind die Zeilen durcheinander geraten. Ordnen Sie die Zeilen so an, dass die Prozedur fehlerfrei funktioniert. (Die Zeilennummerierung dient nur zur Orientierung.)

```
> 1. teilersumme:=proc(n::posint)
   2. i:=i+1;
   3. sum:= 0;
   4. while i <= n do
   5. end:
   6. i:= 1;
   7. if n mod i = 0 then
   8. fi;
   9. sum:=sum+i;
  10. od;
  11. local i, sum;
  12. print(sum);
```

[8 Punkte]

## Aufgabe 8

Schreiben Sie eine Prozedur, die ausgibt, auf wie viele verschiedene Arten eine natürliche Zahl  $n$  als Summe von 3 Quadraten geschrieben werden kann. Die Zahl  $n$  soll der Prozedur übergeben werden.  $0^2$  ist als Summand erlaubt. Verschiedene Kombinationen der Summanden sowie Vorzeichenwechsel werden als **eine** Möglichkeit gezählt.

[12 Punkte]

# Anhang

Maple-Befehl	Erläuterung
#	Kommentar
{}	leere Menge in Maple
[]	leere Liste
!	Fakultät
->	Definition von Abbildungen
<>	ungleich, $\neq$
abs()	Absolutbetrag der übergebenen Zahl
add()	Addition von mehreren Summanden
chrem()	Algorithmus des Chinesischen Restsatzes
conjugate()	komplex konjugierte Zahl
divisors()	Teiler der übergebenen Zahl
do	Beginn der Anweisungen einer Schleife
elif	Teil der if-then-else-Anweisung
else	Teil der if-then-else-Anweisung
end:	Ende der Maple-Prozedur
evalf()	Auswertung von Ausdrücken mit Konstanten/Funktionen
even	gerade ganze Zahl
factorial()	Fakultät der übergebenen Zahl
floor()	größte ganze Zahl, die kleiner oder gleich ist
for	for-Schleife
GaussInt	Gauß'sche Zahlen-Programmbibliothek
I	imaginäre Einheit
if	if-Anweisung
ifactor()	faktoriert die übergebene Zahl
igcd()	Berechnung des ggT der übergebenen Zahlen
igcdex()	Berechnung des ggT mit erweitertem Euklidischen Algorithmus
ilcm()	Berechnung des kgV der übergebenen Zahlen
Im()	Imaginärteil der übergebenen komplexen Zahl
infinity	Befehl für $\infty$
integer	ganze Zahl
iquo()	Quotient bei Division mit Rest
irem()	Rest bei Division mit Rest
isolve()	ganzzahliges Lösen von Gleichungen
isprime()	Primzahltest
ithprime()	$i$ -te Primzahl
local	lokale Variablen

<code>minus</code>	Komplement von Mengen
<code>mod</code>	Rest bei Division mit Rest
<code>mods</code>	symmetrische Form beim modulo-Rechnen
<code>msolve()</code>	modulares Lösen von Gleichungen
<code>mul()</code>	Multiplizieren mehrerer Faktoren
<code>nextprime()</code>	nächstgrößere Primzahl
<code>nops()</code>	Anzahl der Elemente einer Liste
<code>numtheory</code>	Zahlentheorie-Programmbibliothek
<code>od</code>	Ende der Anweisungen einer Schleife
<code>odd</code>	ungerade ganze Zahl
<code>op()</code>	Extraktion der Elemente z.B. einer Liste
<code>phi()</code>	Eulersche $\varphi$ -Funktion
<code>pi()</code>	Primzahlfunktion
<code>plot()</code>	Erstellen von Grafiken
<code>polynom</code>	Polynom
<code>posint</code>	natürliche Zahl
<code>prevprime()</code>	nächstkleinere Primzahl
<code>print()</code>	Ausgabe auf dem Bildschirm
<code>proc()</code>	Maple-Prozedur
<code>product()</code>	Produkt
<code>rand()</code>	Erzeugung einer Pseudozufallszahl
<code>rational</code>	rationale Zahl
<code>Re()</code>	Realteil der übergebenen komplexen Zahl
<code>return()</code>	Beendigung der Prozedur und Speicherung des übergebenen Wertes
<code>round()</code>	Rundung auf nächste ganze Zahl
<code>sigma()</code>	Teilersummenfunktion
<code>sum()</code>	Summe
<code>sum2sqr()</code>	Summe aus zwei Quadraten der übergebenen Zahl
<code>tau()</code>	Teileranzahlfunktion
<code>then</code>	Teil der if-Anweisung
<code>type()</code>	Datentyp der übergebenen Variable
<code>unapply()</code>	Definition von Abbildungen
<code>union</code>	Vereinigung von Mengen in Maple
<code>while</code>	while-Schleife
<code>with(GaussInt)</code>	Aufruf der Programmbibliothek <code>GaussInt</code>
<code>with(numtheory)</code>	Aufruf der Programmbibliothek <code>numtheory</code>