FernUniversität in Hagen

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

Postanschrift: FernUniversität ·58084 Hagen

(Name, Vorname)

(Straße, Nr.)

(PLZ, Wohnort)

FernUniversität in Hagen
FAKULTÄT für Mathematik und Informatik

KLAUSUR zum Kurs Elementare Zahlentheorie mit Maple (01202)
SS 2014

DATUM: 23.08.2014
UHRZEIT: 10.00 - 12.00 Uhr
KLAUSURORT:

Bearbeitungshinweise

(Bitte vor Arbeitsbeginn durchlesen!)

1. Schreiben Sie Ihre Klausur bitte nicht mit Bleistift und nicht mit einem roten Stift.
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.

	Bemerkungen:

Aufgabe	1	2	3	4	5	6	7	8	Summe
Bearbeitet									
max. Punktezahl	10	10	10	10	10	10	10	10	80
erreichte Punktezahl									
Korrektur									

Prüfergebnis/Note	

Elementare Zahlentheorie mit Maple (01202) SS 2014

Klausur am 23.08.2014:

Aufgabenstellungen

Die Lösungen aller Aufgaben müssen Sie begründen.

Aufgabe 1

Seien $a, b, c, d \in \mathbb{Z}$.

(a) Zeigen Sie: Wenn $a \mid b$ und $b \mid a$ gilt, dann gilt $a = b$ oder $a = -b$.

(b) Beweisen oder widerlegen Sie: Aus $a \mid b$ und $a \mid c$ folgt $b \mid c$.

(c) Zeigen Sie: Wenn $a \mid c$ und $c^2 \mid d^2$ gilt, so gilt $a^2 \mid d^2$.

[*3 + 3 + 4 = 10 Punkte*]

Aufgabe 2

Seien p und q zwei aufeinanderfolgende ungerade Primzahlen. Beweisen Sie: Dann ist $p+q$ das Produkt von mindestens drei – nicht notwendig verschiedenen – Primzahlen.

(Hinweis: $p + q = 2 \cdot \frac{p+q}{2}$.)

[*10 Punkte*]

Aufgabe 3

Beweisen Sie mit Hilfe des Kleinen Satzes von Fermat, dass 11 ein Teiler von $5^{72} - 3$ ist.

[*10 Punkte*]

Aufgabe 4

Sei p eine Primzahl und sei $f : \mathbb{N} \to \mathbb{N}_0$ folgendermaßen definiert:

$$f(n) = \begin{cases} 0, & \text{falls } p \mid n, \\ 1, & \text{sonst.} \end{cases}$$

Gilt $f(mn) = f(m)f(n)$ für $m, n \in \mathbb{Z}$ mit $\mathrm{ggT}(m, n) = 1$?

[*10 Punkte*]

Aufgabe 5

(a) Sei $d > 0$ und keine Quadratzahl. Zeigen Sie, dass die Gleichung

$$X^2 - dY^2 = 4$$

immer nicht-triviale Lösungen (x, y) mit $x, y \in \mathbb{Z}$ und $x, y \neq 0$ hat.

(b) Gegeben seien die Pell'sche Gleichung $X^2 - 6Y^2 = 1$ und die zugehörige Fundamentallösung $(a_1, b_1) = (5, 2)$. Berechnen Sie die Lösung (a_2, b_2).

[*5 + 5 = 10 Punkte*]

Aufgabe 6

Finden Sie eine Gauß'sche Zahl $z = a + bi$ mit $a, b \in \mathbb{N}$ und $N(z) = 493$.

(Hinweis: $493 = 17 \cdot 29$.)

[*10 Punkte*]

Aufgabe 7

Eine Vermutung von Lagrange aus dem Jahre 1775 lautet: Jede ungerade Zahl $n > 5$ läßt sich in der Form $p + 2q$ mit Primzahlen p und q darstellen.

Schreiben Sie eine Prozedur `Lagrange()`, die diese Vermutung für alle ungeraden Zahlen $5 < n \leq 75$ überprüft und für jedes n eine Liste der Form $[n, p, q]$ ausgibt.

Der Prozedur sollen keine Parameter übergeben werden!

[*10 Punkte*]

Aufgabe 8

Folgende Maple-Prozedur ist gegeben:

```
>  # Die Zeilennummerierung dient zu Ihrer Orientierung
   1.  zusgesetzt:=proc(a::posint, b::posint, c::posint, d::posint)
   2.  local n;
   3.  n:= a^2 + b^2 + c^2 + d^2;
   4.  if a*b = c*d then
   5.    print(n);
   6.    ifactor(n);
   7.  else
   8.    print(isprime(n));
   9.  fi;
  10.  end:
```

(a) Vollziehen Sie die Arbeitsweise der Prozedur Schritt für Schritt anhand der Eingabe $(3, 4, 2, 6)$ nach.

(b) Wie ändert sich die Ausgabe bei der Eingabe $(3, 4, 2, 5)$ und warum?

(c) Was passiert, wenn die Eingabe $(3, -4, 2, 6)$ lautet? Begründung.

[*4 + 3 + 3 = 10 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()`	faktorisiert 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 ∞
`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

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