

**Programmierung in der Bioinformatik  
Wintersemester 2013  
Übungen zur Vorlesung: Ausgabe am 02.12.2013**

Dieses ist das erste Übungsblatt mit Aufgaben zur Programmiersprache Ruby. Es gibt für Ruby eine umfangreiche Sammlung von Software-Bibliotheken mit vielen Funktionen, die Ihnen beim Lösen der Aufgaben helfen können. Unter <http://ruby-doc.org/core-1.9.3/> finden Sie die entsprechende Dokumentation für die am ZBH installierte Version von Ruby. Diese Version ist auch unter Ubuntu verfügbar.

**Aufgabe 8.1** Sie finden in Stine ein Ruby-Tutorial `ruby20minutes.pdf`, das einige der wichtigsten Konzepte von Ruby einführt. Ein Ausdruck des Tutorials wurde in der Vorlesung verteilt. Bearbeiten Sie dieses Tutorial vor allen anderen Aufgaben, da diese darauf aufbauen. Verwenden Sie zur Bearbeitung die interaktive Ruby-Shell `irb`, die es erlaubt, Fragmente von Ruby-Programmen auszuprobieren.

**Aufgabe 8.2** Implementieren Sie ein Ruby-Skript, das für alle Temperaturen von 1 bis 100 Grad Celsius die entsprechende Temperatur in Fahrenheit ausgibt. Zur Erinnerung: wenn  $t_C$  die Temperatur in Celsius ist, dann ist  $t_F = (t_C * 9) / 5 + 32$  die entsprechende Temperatur in Fahrenheit. Verwenden Sie dazu die Klasse `Range` und die Funktion `each` oder die Methoden `Integer.times` oder `Integer.upto` um über den genannten Temperatur-Bereich zu iterieren.

Beachten Sie, dass Sie eine ganze Zahl `i` zuerst in eine Fließkommazahl umwandeln müssen, damit die Rechenoperationen korrekt funktionieren. Das erreicht man mit der Methode `Integer.to_f`.

Um dieses zu veranschaulichen, werten Sie die folgenden Anweisungen in `irb` aus:

```
print 6 / 4
print 6.0 / 4.0
print 6.to_f / 4.to_f
```

Sie sehen, dass die erste Anweisung das Ergebnis „Division mit Rest“ liefert, und dass die zweite und dritte Zeile das erwartete Ergebnis `1.5` liefert.

Formatieren Sie die Ausgabe mit `IO.printf`. Verwenden Sie dazu die Anweisung

```
printf("c = %.2f f = %.2f\n", celsius, fahrenheit)
```

Vergleichen Sie die Ausgabe Ihres Programms mit der Datei `celsius_out.txt`, die im Material zum Übungsblatt in Stine zur Verfügung steht. Zum Vergleich können Sie die Programme `diff`, `sdiff`, oder `vimdiff` verwenden.

**Aufgabe 8.3** Machen Sie sich mit dem Ruby Interpreter und seinen Fehlermeldungen vertraut. Verwenden Sie dazu den folgenden fehlerhaften Programmtext

```
#!/usr/bin/env ruby

dna = 'ACGAATT\tACTTTAGC'
rna = dna.gsub!(/T/, /U/)
```

```

print "Here "is the starting DNA:\t"
print "#{dna}\n\n"

print "Here is the result:\n\n"
puts "#{rna}\n"
exit 0

```

den Sie im Material zum Übungsblatt in Stine in der Datei `error10-1.rb` finden.

1. Erstellen Sie eine Kopie dieser Datei in der Datei `accept10-1.rb`. Modifizieren Sie diese zunächst so, dass der Ruby-Interpreter keine Fehlermeldungen mehr zeigt. Die Anzahl der Änderungen soll dabei möglichst klein sein.
2. Erstellen Sie in der Datei `correct10-1.rb` eine Kopie von `accept10-1.rb`, und modifizieren `correct10-1.rb` so, dass es die folgende Ausgabe liefert:

```

Here is the starting DNA: ACGAATT ACTTTAGC
Here is the result: ACGAAUU ACUUUAGC

```

Dokumentieren Sie Ihre Änderungen. Dazu können Sie das Programm `sdiff -s` verwenden.

**Aufgabe 8.4** In dieser Aufgabe sollen Sie ein Programm entwickeln, um die Schmelztemperatur einer DNA näherungsweise zu bestimmen. Dazu können Sie für eine DNA-Sequenz  $S$  der Länge  $|S|$  die folgende Formel anwenden:

$$T_m(S) = \begin{cases} 4 \cdot (|\text{occ}(S, G)| + |\text{occ}(S, C)|) + 2 \cdot (|\text{occ}(S, A)| + |\text{occ}(S, T)|) & \text{falls } |S| \leq 14 \\ 64.9 + 41 \cdot (|\text{occ}(S, G)| + |\text{occ}(S, C)| - 16.4)/|S| & \text{sonst} \end{cases}$$

Dabei ist  $|\text{occ}(S, b)|$  die Anzahl der Vorkommen von Base  $b$  in  $S$ .  $T_m(S)$  ist die Schmelztemperatur in Grad Celsius.

Schreiben Sie ein Ruby-Programm, das eine DNA-Sequenz als Parameter auf der Kommandozeile bekommt und die Schmelztemperatur, entsprechend obiger Formel ausgibt. Es sollen nur Eingaben, bestehend aus den Zeichen A, C, G, und T (in Gross- oder Kleinschreibung) akzeptiert werden.

Hinweis: Schauen Sie sich in der Ruby-Dokumentation die Methoden der Klasse `String` an. Es könnte insbesondere die Methode `count` hilfreich sein.

**Die Lösungen zu diesen Aufgaben werden am 16.12.2013 besprochen.**