Betriebssysteme Linux

HA2 Shell-Script 1

FH Burgenland

UNIVERSITY OF APPLIED SCIENCES

BRINGT BESONDERES ZUSAMMEN

DI Franz Knipp
WS 2020/21

Aufgabenbeschreibung

1. **Lesen Sie die allgemeine Aufgabenstellung.**
2. **Lesen Sie die persönliche Aufgabenstellung.**
    1. Endziffer 1 und 6 des PKZ: *Aufgabe 1* Minimum, Maximum, Summe der eingegebenen Argumente
    2. Endziffer 2 und 7 des PKZ: *Aufgabe 2* Entwickeln von Reihen
    3. Endziffer 3 und 8 des PKZ: *Aufgabe 3* Erkennung der Monotonie von Zahlenreihen
    4. Endziffer 4 und 9 des PKZ: *Aufgabe 4* Bilden der Ziffernsumme der Argumente
    5. Endziffer 5 und 0 des PKZ: *Aufgabe 5* Erkennung, welche Argumente Primzahlen sind
3. **Programmieren Sie die persönliche Aufgabenstellung als Bash Script.**
4. **Der Name des Scripts ist in der Aufgabenstellung angegeben.**
5. **Beachten Sie dazu die Hinweise auf den folgenden Folien.**
6. **Geben Sie Ihr Beispiel gemäß der Beschreibung in Moodle ab.**

Allgemeine Aufgabenstellung

- Alle Aufgabenstellungen befassen sich mit **Mathematik**. Verwenden Sie dazu die in der **Bash integrierten Arithmetik-Funktionen**.
- Als **Eingabewerte** für die Aufgaben dienen immer die **Argumente**, die beim Aufruf des Programms übergeben werden.
- Überlegen Sie, welche Argumente gültig sind und führen Sie eine entsprechende **Überprüfung** in Ihrem Programm durch.
- Führen Sie die **Ausgabe der Rechenergebnisse** gemäß Aufgabenstellung durch.

Vorlage für den Programmaufbau

Beachten Sie folgende Punkte:

- Starten Sie das Programm mit einer deutschen **Kurzbeschreibung** des Programms in maximal zwei Zeilen, ausgeführt als Kommentar.
- Geben Sie Ihren Namen als **Autor** an.
- Die Zeilenlänge soll 80 Zeichen nicht überschreiten.
- Rücken Sie Befehlsblöcke ein.
- Verwenden Sie aussagekräftige Variablennamen.

```
#!/bin/bash

# Zeigt alle Argumente an.
#
# Autor: Franz Knipp

# Argumente prüfen

# ...

# Hauptfunktion

for i ; do
  echo $i
done
```

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES

Mathematik in der Bash

Die Bash stellt eine Reihe von **Arithmetikfunktionen** für ganze Zahlen bereit.

Die Funktionen sind an der Syntax **((…))** erkennbar. Die Ausgabe einer Berechnung ist mit **$((…))** möglich.

Der Zugriff auf Argumente erfolgt mit $1, …

Vor Variablennamen ist kein $ erforderlich.

Details:

- `man bash`
- Abschnitt: ARITHMETIC EVALUATION

Beispiele:

```
((x=4))
((x++))
((sum+=$1))
((A=x*y))

if ((sum == 0)); then echo Summe ist 0; fi
echo $((x*y))
```

Grundrechnungsarten:	`+ - * / % **`
Zuweisungen:	`= += -= *= …`
Vergleiche:	`< <= == != >= >`
Bitweise Operatoren:	`! & | ^`

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES

Aufgabe 1: Minimum, Maximum, Summe der eingegebenen Argumente

- **Der Programmname ist *minmaxsum***
- **Das Programm erhält eine Reihe von ganzen Zahlen als Argumente.**
- **Es ermittelt:**
  - Die kleinste Zahl
  - Die größte Zahl
  - Die Summe der Zahlen
- **… und gibt diese gemäß dem folgenden Beispielaufruf aus:**

```
$ minmaxsum 1 9 3 12
1 12 25
```

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES

Aufgabe 2: Entwickeln von Reihen

- **Der Programmname ist *reihe***
- **Das Programm erhält zwei oder drei Zahlen als Argumente: Startwert, Endwert, Intervall (ist 1, wenn nicht angegeben).**
- **Es gibt die Zahl x, das Quadrat x^2 und dritte Potenz x^3 aus:**

```
$ reihe 2 4
2 4 8
3 9 27
4 16 64
```

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES

Aufgabe 3: Erkennung der Monotonie von Zahlenreihen

- **Der Programmname ist *monotonie***
- **Das Programm erhält eine Zahlenfolge in den Argumenten und muss erkennen, ob die Zahlenfolge eine der folgenden Eigenschaften hat:**
  - Streng monoton steigend, Monoton steigend, Konstant, Monoton fallend, Streng monoton fallend
  - Falls keine der Eigenschaften zutrifft: Keine Monotonie erkannt
- **Beispiel:**

```
$ monotonie 1 3 5 9
Streng monoton steigend
$ monotonie -4 8 2 9
Keine Monotonie erkannt
```

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES

Aufgabe 4: Bilden der Ziffernsumme der Argumente

- **Der Programmname ist *ziffernsumme***
- **Das Programm erhält eine oder mehrere Zahlen als Argumente und berechnet für jede angegebene Zahl die Ziffernsumme.**
- **Jede Zeile der Ausgabe enthält die Zahl und deren Ziffernsumme.**
- **Beispiel:**

```
$ ziffernsumme 29 115 2001
29 11
115 7
2001 3
```

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES

Aufgabe 5: Erkennung, welche Argumente Primzahlen sind

- **Der Programmname ist *primzahlen***
- **Das Programm erhält eine oder mehrere Zahlen als Argumente und bestimmt für jede angegebene Zahl, ob sie prim ist.**
  - Eine Primzahl ist eine Zahl, die nur durch sich selbst und durch 1 ohne Rest geteilt werden kann.
  - Den Divisionsrest bestimmen Sie mit dem Operator %, z. B. `((14 % 5))` ergibt 4
  - Eine Primzahl wird im Ergebnis mit 1 markiert, andere Zahlen mit 0.
- **Beispiel:**

```
$ primzahlen 29 33
29 1
33 0
```

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES

Bewertungsraster

Bewertungsgrundlage	Anteil
Korrektheit Ist das Programm lauffähig? Ist das Ergebnis des Programms korrekt?	**40%**
Vollständigkeit Überprüft das Programm die Argumente auf Anzahl und Korrektheit? Gibt es entsprechende Fehlermeldungen aus?	**30%**
Form Enthält das Programm Kurzbeschreibung und Autor als Kommentar? Ist das Programm gut lesbar (Einrückungen, Zeilenlänge)? Ist der Name korrekt? Enthält das Programm eine Hilfefunktion?	**30%**

FH Burgenland
UNIVERSITY OF APPLIED SCIENCES