**GL der Programmierung**

**Schleifen**

**Hausübung der 5. Einheit vom 23. 4. 2018**

**! Achtung !  
Lösen Sie bitte mindestens 6 Beispiele.**

**Die Zahlen in eckigen Klammern geben die Punkte für die Präsentation der richtigen Lösung an.  
Benennen Sie die Beispiele wie in der Angabe angegeben.   
Geben Sie alle in dasselbe Package und benennen Sie das Package mit Ihrem Namen (Nachname + Vorname mit Kleinbuchstaben beginnend) und der Nummer der Hausübung, getrennt durch ‘\_‘, also z.B.**

**mayerRobert\_5**

**Exportieren Sie das Package aus eclipse – vergessen Sie dabei nicht das Package zu selektieren und „Create only selected directories“ zu wählen – in eine Datei, die Ihren Namen, die Nummer der Hausübung, sowie die Nummern der gelösten Beispiele trägt, also z.B.:**

**Mayer\_Robert\_Aufgabe\_5\_Bsps\_1\_3\_4  
Geben Sie die exportierte Datei rechtzeitig auf moodle ab.**

## Beispielgruppe 5 – Schleifen

Lesen Sie die Werte für die folgenden Beispiele stets im main()-Programm ein.   
Führen Sie die Berechnung bei den mit einem Stern (\*) bei den Punkten gekennzeichneten Beispielen in einem Unterprogramm durch.

1. Radien: Schreiben Sie ein Programm, bei welchem der Radius eines Kreises eingegeben und daraus dessen Umfang berechnet wird. Dies soll so lange wiederholt werden, bis der Benutzer den Radius 0 eingibt. Bei Eingabe einer negativen Zahl soll eine entsprechende Meldung ausgegeben werden. [4]
2. RechteckSchleife: Lesen Sie die Länge und Breite eines Rechtecks ein. Fragen Sie danach den Benutzer, was berechnet werden soll.   
   Durch die Eingabe von

1 soll der Umfang  
2 soll die Fläche

3 soll die Länge der Diagonale

des Rechtecks berechnet werden.

Durch die Eingabe von

4 soll die Länge und Breite eines weiteren Rechtecks eingelesen werden.

Dies wird so lange wiederholt, bis durch die Eingabe von 0 das Programm beendet wird. (Bei einer anderen Eingabe als oben angegeben soll eine entsprechende Meldung ausgegeben werden). [5]

1. Fakultät: Berechnen Sie die Fakultät von n mit Hilfe einer Schleife. [3] \*
2. Ziffernsumme: Schreiben Sie ein Programm, das eine ganze Zahl einliest und mit Hilfe der Ziffernsumme überprüft, ob die eingegebene Zahl durch 3 teilbar ist. [5] \*
3. EinMalEins: Geben Sie das kleine 1 mal 1 aus. [3]
4. Intervall: Lesen Sie 2 Intervallgrenzen und danach beliebig viele Zahlen (Abschluss mit 0) ein, und geben Sie aus, wie viele Zahlen kleiner als die untere Intervallgrenze, wie viele im Intervall und wie viele größer als die obere Intervallgrenze waren.   
   (Die Zahlen müssen nicht aufgehoben werden). [5]
5. Folge: Überprüfen Sie, ob die Folge  
   ni+1 = ni / 2 für gerade ni  
   ni+1 = 3ni + 1 für ungerade ni  
   für unterschiedliche (beliebige) positive Startwerte n1 immer den Wert 1 erreicht. [6]
6. Primzahl: Bestimmen Sie, ob eine eingegebene natürliche Zahl eine Primzahl ist. [3] \*
7. AllePrimzahlen: Lesen Sie eine Zahl ein und geben Sie alle Primzahlen aus, die kleiner als diese Zahl sind. [4]  
   Beispiel:  
   Eingabe: 10  
   Ausgabe: 2, 3, 5, 7   
   [Anmerkung: Die Ausgabe kann auch ohne Beistriche in mehreren Zeilen erfolgen.]
8. Statistik: Lesen Sie beliebig viele Zahlen (Abschluss mit 0) ein, und geben Sie das Maximum, das Minimum und den Mittelwert der eingegebenen Werte aus. [5]
9. Zweierpotenz: Schreiben Sie ein Programm, das eine natürliche Zahl einliest und die größte Zweierpotenz ausgibt, die die eingelesene Zahl teilt. [6]

|  |  |
| --- | --- |
| Eingabe | Ausgabe |
| 17 | 1 |
| 48 | 16 |
| 1024 | 1024 |

1. PerfekteZahl: Eine Zahl heißt perfekt, wenn die Summe ihrer echten Teiler (das sind alle Teiler, die kleiner als die Zahl selbst sind) gleich groß ist, wie die Zahl. Zum Beispiel ist 6 eine perfekte Zahl, da 1, 2 und 3 die echten Teiler von 6 sind und 1+2+3=6 gilt. Falls die Summe der echten Teiler kleiner ist als die Zahl, heißt die Zahl defizient. Falls die Summe der echten Teiler größer ist als die Zahl, heißt die Zahl abundant. Schreiben Sie ein Programm, das eine natürliche Zahl einliest und ausgibt, ob die eingelesene Zahl perfekt, defizient oder abundant ist. [8] \*

|  |  |  |
| --- | --- | --- |
| Eingabe | Ausgabe | Begründung |
| 5 | defizient | 1 ist einziger echter Teiler |
| 6 | perfekt | 1+2+3=6 |
| 12 | abundant | 1+2+3+4+6=16>12 |

1. GGTBruteForce: Lesen Sie zwei Zahlen ein und berechnen Sie den größten gemeinsamen Teiler (ggT) dieser Zahlen mit der Brute Force Methode. [3] \*
2. GGTEuklid: Lesen Sie zwei Zahlen ein und berechnen Sie den größten gemeinsamen Teiler (ggT) dieser Zahlen mit dem klassischen Algorithmus von Euklid. [3] \*
3. kgV: Lesen Sie zwei Zahlen ein und berechnen Sie das kleinste gemeinsame Vielfache (kgV) dieser Zahlen. [7] \*

Beispiele:

Das kgV von 15 und 25 ist 75   
(75 mod 15 = 0, 75 mod 25 = 0, keine kleinere Zahl als 75 hat diese Eigenschaft)   
Das kgV von 8 und 12 ist 24

(24 mod 8 = 0, 24 mod 12 = 0, keine kleinere Zahl als 24 hat diese Eigenschaft)

1. Akummuliert: Schreiben Sie ein Programm, welches eine Zahl x einliest und danach so lange die Zahlen von 1 bis n aufsummiert bis die Summe die Zahl x übersteigt. Geben Sie sowohl n, als auch die akummulierte Summe aus. [6]  
   Beispiel:

Eingabe: 9  
Ausgabe: Zahl: 4; Summe: 10  
Erklärung:

1 < 9

1 + 2 = 3 < 9

1 + 2 + 3 = 6 < 9

1 + 2 + 3 + 4 = 10 > 9

1. Kugelkorb: In einem Korb liegen 66 verschiedenfarbige Kugeln:   
   Es sind doppelt so viel weiße wie grüne, eine blaue weniger als grüne und sieben rote mehr als grüne.  
   Wie viele Kugeln von jeder Farbe sind im Korb? [4]
2. HerrHuber: Herr Huber las in den letzten 4 Jahren 43 Bücher.  
   Das Produkt seiner in diesen 4 Jahren pro Jahr gelesenen Bücher beträgt 6720.  
   Er ist stolz darauf, dass er seinen Bücherkonsum von Jahr zu Jahr gesteigert hat, und dass er im 4. Jahr 5-mal so viele Bücher las wie im ersten Jahr.  
   Berechnen Sie die Anzahl der von Herrn Huber pro Jahr gelesenen Bücher. [5]
3. Schnecke: Eine Schnecke sitzt vor einer 4,5 Meter hohen Mauer. Jeden Tag klettert sie 50 cm nach oben, in jeder Nacht rutscht sie 10% ihrer momentanen Gesamthöhe nach unten. An welchem Tag ist sie oben? [7]
4. AlternierendeReihe: Berechnen Sie die Reihe

1 – ½ + 1/3 – ¼ + 1/5 - ...

Bis zum 100. Glied (1 - ... - 1/100) und geben Sie dieses aus. [7]

1. Groesser10: Ab welchem n ist die (Summe der) Reihe

1 + ½ + 1/3 + ¼ + ... + 1/n

größer als 10? [7]

1. Aufsteigend: Lesen Sie beliebig viele Zahlen ein (Abschluss mit 0, Sie brauchen diese nicht zu speichern). Geben Sie an, ob die Zahlen aufsteigend sortiert sind. [5]  
   Beispiele:

1, 3, 6, 8, 13, 0 – aufsteigend sortiert

2, 5, 7, 5, 8, 9, 0 – nicht aufsteigend sortiert (5 kommt nach 7, ist aber kleiner als 7)

1. LaengsteFolge: Lesen Sie beliebig viele Zahlen ein (Abschluss mit 0, Sie brauchen diese nicht zu speichern). Geben Sie die Länge der längsten Folge von aufsteigend sortierten Zahlen an.  
   Beispiel: [6]

3, 4, 8, 2, 4, 3, 6, 5, 0 – aufsteigende Teilfolgen:

3, 4, 8 – 2 ist wieder kleiner

2, 4 – 3 ist wieder kleiner

3, 6 – 5 ist wieder kleiner

Längste Teilfolge hat 3 Elemente

1. Intervall2: Lesen Sie zwei ganze Zahlen ein. [4]
   * Falls die zweite Zahl kleiner ist als die erste, geben Sie nichts aus.
   * Andernfalls geben Sie alle ganzen Zahlen von der ersten bis zur zweiten aus.

Beispiel:

Eingabe: 13 und 11  
 Ausgabe: *nichts*

Eingabe: 7 und 11

Ausgabe; 7, 8, 9, 10, 11

1. NMalN: Lesen Sie eine Zahl (n) ein und berechnen Sie mit einer Schleife: [5] \*

1 \* n + 2 \* n + 3 \* n + ... + n \* n

1. NHochN: Lesen Sie eine Zahl (n) ein und berechnen Sie:

n  
∑ ii  
i=1

(also 11+22+33+...+nn oder 1 + 2 \* 2 + 3 \* 3 \* 3 + ... + n \* n \* ... \* n)

ohne die Potenzfunktion zu benutzen. [6] \*