



**THU**

**Technische  
Hochschule  
Ulm**

Vorkurs Programmieren - Aufgaben

# Variablen und Ein-/Ausgabe

## Aufgabe 1 ~ Willkommen Fremder!

Schwierigkeit: ★

Am Anfang steht immer eine Begrüßung! Auch beim Programmiervorkurs. Erstellen Sie daher eine neue Java Klasse und geben Sie dieser einen Namen (z.B. Greetings). Falls in ihrer Klasse noch keine `main()` Methode existiert, fügen Sie diese hinzu. Ihre Klasse sollte nun wie folgt aussehen:

```
public class Greetings {  
    public static void main(String[] args) {  
  
    }  
}
```

Schreiben Sie nun ein Programm, welches zunächst nach dem Namen des Benutzers fragt und diesen einliest. Anschließend soll der Name für eine Begrüßung benutzt werden. Die Ausgabe könnte dabei wie folgt aussehen:

```
Hello MyWunderfulName !
```

## Aufgabe 2 ~ Java und seine Datentypen

Schwierigkeit: ★

Welche Datentypen sind Ihnen bekannt? Geben Sie jeweils mindestens ein Beispiel in Java an.

Datentyp	Beispiel in Java
Integer	int number = 12;
...	



Sie können diese Aufgabe auf einem Block oder am Rechner mit einem beliebigen Editor lösen

## Aufgabe 3 ~ Was für ein Typ bist du?

Schwierigkeit: ★

Finden Sie für folgende Verwendungszwecke geeignete Datentypen. Es kann für jeden Verwendungszweck mehrere Datentypen geben.

Verwendungszweck	Beispiel in Java
Alter eines Menschen	int alter = 21;

Verwendungszweck	Beispiel in Java
Das Jahresgehalt in ganzen Euro-Beträgen	
Das Geschlecht einer Person	
Die Anrede einer Person (Herr, Frau, ...)	
Die eulerische Zahl e mit 14 Nachkommastellen	
Das Gewicht in kg	



Sie können diese Aufgabe auf einem Block oder am Rechner mit einem beliebigen Editor lösen

## Aufgabe 4 ~ Deklaration und Initialisierung

Schwierigkeit: ★

Im folgenden Programm werden Variablen deklariert und initialisiert. Geben Sie an, welche Variablen es gibt und in welcher Zeile sie deklariert und initialisiert werden.

```
public class DeclarationInit {
    public static void main(String[] args) {
        int number;
        String word;
        double realNumber;

        number = 1;
        word = "Hello";
        realNumber = 0.2;
        System.out.println(number);
        System.out.println(word);
        System.out.println(realNumber);

        int anotherNumber = 5;
        String anotherWord = "You";
        double anotherRealNumber = 8.15;
        System.out.println(anotherNumber);
        System.out.println(anotherWord);
        System.out.println(anotherRealNumber);
    }
}
```

## Aufgabe 5 ~ Selbst gemacht

Sie haben Oma's Apfelkuchenrezept gefunden, jedoch sind die Mengenangaben nicht mehr vorhanden. Doch dieses Problem lässt sich einfach beheben, den Oma hat zum Glück ein paar Kommentare hinterlassen. Sie müssen nur noch die nötigen Variablen deklarieren und ihnen den passenden Wert zuweisen. Wie lautet Oma's Rezept?

```

public class PieReceipt {
    public static void main(String[] args) {
        System.out.println("Apple Pie Receipt:");
        System.out.println("=====");

        // declare variable "eggs" of type "int"

        // assign value 4 to "eggs"

        System.out.println("- " + eggs + " Eggs");

        // declare variable "milk" of type "double"
        // and assign value 100 in one line

        System.out.println("- " + milk + " mL Milk" );

        // declare variable "flour" of type "double"

        // assign value 200 to "flour"

        // add 150 to "flour"

        System.out.println("- " + flour + " mg Flour");

        // declare variable "apple" of type "int"
        // assign the value of "eggs" to "apple"

        System.out.println("- " + apple + " Apples");
    }
}

```

## Aufgabe 6 ~ Eine runde Sache

Schwierigkeit: ★

Schreiben Sie ein Programm, welches den Umfang und die Fläche eines Kreises berechnen kann. Die berechneten Werte sollen anschließend ausgegeben werden. Für die Berechnung selbst soll nur der Radius des Kreises eingelesen werden.

- Formel für den Kreisumfang:  $U = 2 * \text{PI} * r$
- Formel für die Kreisfläche:  $F = 2 * \text{PI} * r^2$

Sie können dabei wie folgt vorgehen:

1. Definieren Sie eine Variable `PI` mit dem Wert 3.14159
2. Definieren Sie eine Variable `radius` und lesen sie einen Wert mit dem Scanner ein
3. Berechnen Sie sowohl den Umfang als auch die Fläche und speicher Sie die Werte in Variablen

4. Geben Sie die Werte mit Hilfe von *System.out* aus



Das Zeichen  $\wedge$  steht in Java nicht für Potenz. Wie kann  $r^2$  alternativ berechnet werden?

## Aufgabe 7 ~ Error: Can not compute!

Schwierigkeit: ★

In ihrem Projekt hat ihr Kollege ein Programm geschrieben. Dieses soll Umfang, Grundfläche, Mantelfläche, Oberfläche und Volumen eines Zylinders berechnen. Leider sind in dem Programm einige Fehler enthalten. Können Sie ihrem Kollegen helfen die Fehler (8) zu finden?

```
import java.util.Scanner;

public class ErrorCode {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.out);
        double pi = 3,1415;

        System.out.print("Radius: ");
        double radius = scan.nextDouble();

        System.out.print("Height: ")
        double height = scan.nextDouble();

        double circumference = 2 * pi * -(-radius);
        double baseArea = pi * radius ^ 2;
        double lateralSurfaceArea = circumference * height;
        double surface = 2 * baseArea + lateralSurfaceArea;
        double volume = baseArea * height;

        System.out.println("Circumference: + circumference");
        System.out.println("Base Area: " + baseArea);
        System.out.println("Lateral Surface Area: "+lateralSurfaceArea);
        System.out.println("Surface: " surface);
        System.out.println("Volume: "+volume);
    }
}
```

## Verzweigungen

### Aufgabe 1 ~ Völlig logisch!

Schwierigkeit: ★

Folgende Variablen sind definiert:

```

int a = 4;
int b = 3;
int c = 3;
double d = 4.5;
int e = 1;

```

Welche der folgenden Aussagen sind wahr (*true*) und welche sind falsch (*false*)?

Ausdruck	Ergebnis
<code>a != 4</code>	
<code>a &gt; b</code>	
<code>b == c</code>	
<code>(b+1) == 4</code>	
<code>(a/b) == 1</code>	
<code>(b/a) &gt; 0</code>	
<code>(a%e) != 0</code>	
<code>(a &gt; 0) &amp;&amp; (a &lt;= 4)</code>	
<code>!(e &lt; c)</code>	

## Aufgabe 2 ~ Größer, kleiner, gleich

Schwierigkeit: ★

Schreiben Sie ein Programm, bei welchem Sie zwei Zahlen einlesen können. Ihr Programm soll anschließend ausgeben, welche der beiden Zahlen die größte und welche die kleinste Zahl ist. Falls gleiche Zahlen eingegeben worden sind, so soll diese Zahl ebenfalls ausgegeben werden.

## Aufgabe 3 ~ Body-Mass-Index

Schwierigkeit: ★

Für die Berechnung des Body-Mass-Index (BMI) sollen Sie ein Programm schreiben. Der Body-Mass-Index dient als Maß zur generellen Beurteilung des Körpergewichts. Der Index gibt das Verhältnis des Körpergewichts  $G$  (kg) zu der Größe  $h$  (m) an. Er kann mit der folgenden Formel berechnet werden:

- $BMI = G / h^2$

Die Beurteilung folgt dabei der nachstehenden Tabelle

BMI	Beurteilung
$BMI < 20$	Untergewicht
$20 \leq BMI < 25$	Normalgewicht
$25 \leq BMI < 30$	Übergewicht

BMI	Beurteilung
$30 <= \text{BMI} < 40$	Stakes Übergewicht
$\text{BMI} >= 40$	Extremes Übergewicht

Ihr Programmausgabe könnte wie folgt aussehen:

```
Weight (kg): 55
Height (m): 1.6
BMI = 21.48437499999 > You have normal weight!
```

## Aufgabe 4 ~ 20% auf alles!

Schwierigkeit: ★

Eine Firma die Tiernahrung verkauft hat Sie gebeten eine Software zu schreiben, welche den passenden Mengenrabatt bei einer Bestellung berechnet. Ab 10kg soll es einen Rabatt von 10% geben und ab 50kg von 20%. Schreiben Sie ein Programm, welches zunächst den Preis pro Kilogramm und danach die Bestellmenge einliest. Danach soll das Programm den Preis ohne Rabatt, mit Rabatt und die Differenz ausgeben.

## Aufgabe 5 ~ Alle Jahre wieder

Schwierigkeit: ★★

Sie wollen wissen, wann das nächste Schaltjahr ist! Zum Glück gibt es eine Möglichkeit dieses vom Computer berechnen zu lassen. Sie müssen es nur noch programmieren. Ihr Programm soll hierbei zunächst das Jahr einlesen, welches überprüft werden soll. Anschließend wird dieses getestet.

*Ein Schaltjahr erfüllt folgende Bedingungen*

- Es ist ein Schaltjahr, wenn die Jahreszahl durch 4 teilbar ist
- Ist es auch ganzzahlig durch 100 teilbar, so ist es kein Schaltjahr, außer ...
  - ... das Jahr ist ebenfalls ganzzahlig durch 400 teilbar

Folgende Jahre sind Schaltjahre:

1808	1904	2000	2112	2244	2332	2380	2400
------	------	------	------	------	------	------	------



Ob eine Zahl ganzzahlig durch eine andere Zahl teilbar ist kann mit dem Modulo-Operator herausgefunden werden.

```
if(year % 4 == 0) {
    //...
}
```

# Schleifen

## Aufgabe 1 ~ Wieder und wieder und wieder ...

Schwierigkeit: ★

Zu Beginn soll ein einfaches Programm geschrieben werden, welches einen beliebigen Satz wiederholt. Der Nutzer soll hierbei sowohl den Satz, als auch die Anzahl der Wiederholungen festlegen können. Schreiben Sie die Wiederholung mit allen ihnen bekannten Schleifen.

## Aufgabe 2 ~ Dreimal darfst du raten!

Schwierigkeit: ★★

Schreiben Sie ein kleines Spiel, bei welchem der Spieler eine vom Computer zufällige Zahl erraten muss. Der Spieler hat dabei drei Versuche. Die Zufallszahl soll zwischen 0 und 100 sein. Falls der Spieler falsch liegt, wird ausgegeben ob die Zufallszahl größer oder kleiner als die eingegebene Zahl ist. Der Ablauf des Spiels ist wie folgt:

*Spielablauf*

1. Die Zufallszahl wird bestimmt
2. Der Benutzer soll eine Zahl eingeben
3. Wenn die Zahl falsch ist, einen entsprechenden Hinweis ausgeben
  - a. Ansonsten ausgeben das der Spieler gewonnen hat und Spiel beenden
4. Falls der Spieler seine drei Versuche aufgebraucht hat wird das Spiel beendet



Eine Zufallszahl können Sie mit folgenden Anweisung berechnen lassen:

```
import java.util.Random;
//....
public static void main(String[] args) {
    Random rand = new Random();
    int randomNumber = rand.nextInt(101);
}
```

## Aufgabe 3 ~ Von Rauten und Rechtecken

Schwierigkeit: ★★

Ihr bester Freund mag Rauten und Rechtecke. Er bittet Sie ein Programm für ihn zu schreiben, welches ein Rechteck aus Rauten ausgibt. Dabei soll die Breite als auch die Höhe des Rechtecks zuvor eingelesen werden können. Eine Beispielausgabe könnte wie folgt aussehen:

```
Width of rectangle: 5
Height of rectangle: 4
#####
#####
#####
#####
```



Sie brauchen hierfür eine verschachtelte Schleife. Das Grundgerüst könnte wie folgt aussehen:

```
for(int y=0; y<height; y++) {
    for(int x=0; x<width; x++) {
    }
}
```

## Aufgabe 4 ~ Wörter in Ketten

Schwierigkeit: ★★★

Schreiben Sie ein kleines Spiel, bei welchem zwei Spieler Wortketten bilden müssen. Bei diesem Spiel fängt der erste Spieler an mit einem beliebigen Wort. Der zweite Spieler muss darauf ein Wort nennen, dessen Anfangsbuchstabe der Endbuchstabe des letzten genannten Wortes ist. Anschließend ist wieder der erste Spieler dran. Jeder Spieler bekommt dabei die Länge seines Wortes als Punkte gutgeschrieben. Sollte ein Spieler ein Wort nennen, das die Bedingung nicht erfüllt, verliert er alle seine Punkte und hat verloren. Gewonnen hat der erste Spieler mit insgesamt 30 Punkten bzw. mit den meisten Punkten.



Mit Hilfe der Methode `charAt( index )` der Klasse `String` können Sie auf einzelne Buchstaben eines Strings zugreifen. Der erste Buchstabe hat hierbei den Index 0.



Sie können ein Wort mit der Methode `toLowerCase()` umwandeln, sodass es nur aus Kleinbuchstaben besteht.



Mit Hilfe der Methode `length()` der Klasse `String` können Sie die Länge des Strings ermitteln.

```
String text = "Hello";
text.charAt(1); // 'e'
text.toLowerCase(); // "hello"
text.length(); // 5
```

# Aufgabe 5 ~ Musterbeispiel

Schwierigkeit: ★★★

Schreiben Sie ein Programm welches auf der Konsole das Muster `.:|:..|` wiederholt ausgibt. Die genaue Anzahl an Zeichen die Ausgegeben werden soll wird über die Tastatur eingelesen.

```
Amount tokens: 7
.:|:..|

Amount tokens: 21
.:|:..|:..|:..|:..|:..|:..

Amount tokens: 2
.:
```



Mit dem Modulo-Operator und einer Schleife können Sie eine sich wiederholende Kette an Zahlen erzeugen.

## Extra ~ Wrappy Words

Schwierigkeit: ★★★

Anstatt einer Anzahl von Zeichen einzulesen soll nun ein Wort eingelesen werden. Dieses Wort soll anschließend in das Muster eingefügt werden. Eine mögliche Ausgabe könnte wie folgt aussehen:

```
Word to wrap: Hello
.:H:..e:..l:..l:..o:..
```

# Arrays

## Aufgabe 1 ~ Zugriff!

Schwierigkeit: ★

In der nachfolgenden Tabelle sind Zugriffe auf ein Array aufgelistet. Die verwendeten Arrays sind über der Tabelle. Geben Sie an, auf welchen Wert jeweils zugegriffen wird. Aber vorsicht! Manchmal ist ein Zugriff nicht möglich!

```
int[] arrayOne = new Int[3];
arrayOne[0] = 5;
arrayOne[1] = 2;
arrayOne[2] = arrayOne[0] * 3;

int[] arrayTwo = { 2, 4, 6, 8, 10 };
```

Arrayzugriff	Wert
arrayOne.length	
arrayTwo.length	
arrayOne[1]	
arrayTwo[3]	
arrayTwo[arrayOne.length]	
arrayOne[arrayOne.length - 1]	
arrayTwo[10]	

## Aufgabe 2 ~ Shopping

Schwierigkeit: ★

Sie gehen mal wieder einkaufen und wissen nicht, ob sie das alles überhaupt bezahlen können. Dazu müssten Sie den Wert ihres Einkaufs berechnen. Sie sind aber ziemlich rechenfaul, weshalb der Rechner wieder ran soll. Sie schreiben also ein Programm, welches zuerst wissen möchte, wie viele Werte eingelesen werden. Anschließend können Sie die Werte ihrer Waren in ein Array eingeben und das Programm berechnet Ihnen die Summe.



Sie können auch den Wert einer Variable dazu benutzen, ein Array einer bestimmten Größe zu erstellen.

```
int amount = scan.nextInt();
double[] values = new double[amount];
```

## Aufgabe 3 ~ Ein guter Tausch

Schwierigkeit: ★★

Sie haben folgendes Array gegeben:

```
int myNumbers = { 1, 2, 3, 4, 5 };
```

Schreiben Sie ein Programm, welches jede Zahl mit seinem rechten Nachbarn vertauscht. Die erste Zahl soll dabei am Ende an der letzten Stelle stehen. Benötigen Sie hierfür eventuell die Hilfe von

Variablen? Versuchen Sie das Problem mit möglichst wenig Variablen zu lösen!

## Aufgabe 4 ~ Hab dich gefunden!

Schwierigkeit: ★★

Schreiben Sie ein Programm, welches zunächst eine beliebige Anzahl an Zahlen einliest und in ein Array speichert. Anschließend soll eine Zahl eingegeben werden können, die in dem Array gesucht werden soll. Falls sich diese Zahl in dem Array befindet, geben Sie eine entsprechende Meldung aus. Ansonsten darf der Benutzer eine neue Zahl eingeben, welche wieder im Array gesucht wird. Sobald eine Zahl gefunden wurde, wird das Programm beendet.

## Aufgabe 5 ~ So eine Unordnung!

Schwierigkeit: ★★★

Unglaublich! Sie waren nur kurz weg und schon hat jemand sämtliche Zahlen in ihrem Array vertauscht. Dabei ist das Ordnen der Zahlen immer so zeitaufwendig. Sie beschließen ein Programm zu schreiben, was Ihnen diese nervige Arbeit abnimmt. Das Programm soll hierbei das Array durchlaufen und die jeweils nebeneinander liegenden Zahlen miteinander vergleichen. Sollte die kleinere Zahl an der Stelle mit dem größeren Index stehen, so werden die beiden Zahlen getauscht. Am Ende soll das Array aufsteigend sortiert sein. Zum überprüfen können sie das Array ausgeben.

Ihr durcheinander gebrachtes Array sieht wie folgt aus:

```
int[] myNumbers = { 9, 3, 2, 8, 6, 7, 5, 4, 1 };
```

## Aufgabe 6 ~ Gutes Wetter!

Schwierigkeit: ★★★

Sie haben eine Wetterstation, die Ihnen die Temperatur der letzten 14 Tage zurückliefert. Diese Werte wollen Sie nun mit Hilfe eines Programms näher untersuchen. In der unten stehenden Tabelle finden Sie die aufgezeichneten Temperaturen.

Tag	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Temperatur	12	14	10	7	11	13	12	15	15	18	16	13	10	12

1. Speichern Sie die Temperaturen mit einem geeigneten Datentyp ab.
2. Schreiben Sie nun ein Programm, welches die Tage und Temperaturen in einer Tabelle ausgibt.
3. Erweitern Sie ihr Programm so, die beiden aufeinanderfolgenden Tage ausgegeben werden, welche den größten Temperaturunterschied haben.

# Methoden

## Aufgabe 1 ~ Statistisch gesehen!

Schwierigkeit: ★★

Für eine Statistiksoftware sollen Methoden geschrieben werden, welche es ermöglichen das Minimum, Maximum sowie den Mittelwert aus einem Array an Zahlen zu berechnen. Die Methoden sollen jeweils ein Array mit Zahlen übergeben bekommen und aus diesem Array den entsprechenden Wert zurückgeben. Der Benutzer soll entscheiden können, wie viele Zahlen er eingeben möchte. Anschließend kann er entsprechend viele Zahlen auch eingeben und bekommt am Ende das Minimum, Maximum und den Mittelwert ausgegeben.

```
public static double minimum(double[] numbers) {  
}  
  
public static double maximum(double[] numbers) {  
}  
  
public static double mean(double[] numbers) {  
}
```

## Aufgabe 2 ~ Rückwärts gelesen macht das Sinn!

Schwierigkeit: ★★

Ein Deutschstudent kommt zu ihnen mit einem Problem: Er muss eine Liste an Wörter nach Palindrome untersuchen. Sie sollen für ihn eine Methode schreiben, welche ein Wort übergeben bekommt und dieses testet. Sollte das übergebene Wort ein Palindrom sein, so soll die Methode *true* zurückliefern. Helfen Sie dem armen Deutschstudenten mit seinem unlösbaren Problem und implementieren Sie diese Methode. Sie können ihre Methode mit der folgenden Liste testen:

- Annasusanna
- Dreieck
- Bruggrub
- handnah
- Palindrom
- Malajalam
- Apfelsaft

Sie können folgende Methode als Vorlage benutzen.

```
public static boolean isPalindrom(String text) {  
}
```

## Aufgabe 3 ~ Oh Tannenbaum!

Schwierigkeit: ★★★

Da schon bald wieder Weihnachten ist, schreiben Sie ein Programm welches einen Tannenbaum auf der Konsole ausgeben kann. Hierfür benötigen Sie zwei Methoden. Die erste Methode soll die Krone des Baums ausgeben, die zweite Methode den Stamm. Der Benutzer soll zudem angeben können, wie hoch die Krone des Baums sein soll. Der Stamm des Tannenbaums ist hierbei ein Drittel so hoch wie die Krone. Zudem ist der Stamm auch nur ein Drittel so breit wie die breiteste Stelle der Baumkrone. Für ein symmetrisches Bild muss die Breite des Stamms auch immer eine ungerade Zahl sein.

Die Ausgabe kann beispielsweise so aussehen:

```
Crown Height: 7
  #
 ###
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

Für ihr Programm können Sie folgende Methoden als Vorlage benutzen.

```
public static void printCrown(int height) {
}

public static void printStem(int height, int width, int gap) {
}
```

## Aufgabe 4 ~ Caesar's Buchstabenschubsen

Schwierigkeit: ★★★★★

Sie haben eine Kiste aus dem alten Rom gefunden. Darin befinden sich ein alter Brief eines römischen Feldherrns. Dummerweise ist die Nachrichten verschlüsselt aufgeschrieben. Da der Kiste keine Chiffrierscheibe beilag, brauchen Sie ein Programm um den Text zu entschlüsseln. Von einem Professor der Archäologie aus Amerika wissen Sie das Vorgehen der Verschlüsselung. Jeder Buchstabe des Alphabets wird durch einen Buchstaben ersetzt, welcher 13 Stellen davor oder dahinter liegt. In der folgenden Tabelle ist aufgezeigt, welcher Buchstabe mit welchem anderen Buchstaben verwechselt wird.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Als Grundgerüst kann folgender Code dienen:

```
public static String encrypt(String text) {  
  
}
```

In dem Brief steht folgende Nachricht:

Jrvq uvagra, uvagre qra Jbegoretra, srea qre Yäaqre Ibxnyvra haq Xbafbanagvra yrora qvr Oyvaqgrkgr. Notrfpuvrqra jbuara fvr va Ohpufgnounhfra na qre Xüfgr qrf Frznagvx, rvarf tebßra Fcenpubmrnaf. Rva xyrvarf Oäpuyrva anzraf Qhqra syvrßg qhepu vuera Beg haq irefbetg fvr zvg qra aögvttra Ertryvnyvra. Rf vfg rva cnenqvrzfzngvfpurf Ynaq, va qrz rvarz troengrar Fngmgrvyr va qra Zhaq syvrtra. Avpug rvazny iba qre nyyzäpugvttra Vagrechaxgvba jreqra qvr Oyvaqgrkgr orureefpug – rva trenqrmh habegubtencuvfpurf Yrora. Rvarf Gntrf nore orfpuybß rvar xyrvar Mrvyr Oyvaqgrkg, vue Anzr jne Yberz Vcfhz, uvanhf mh trura va qvr jrvgr Tenzngvx. Qre tebßr Bkzkb evrg vue qniba no, qn rf qbeg jvzzryr iba oöfra Xbzzngn, jvyqra Sentrmrvpura haq uvagreuäygvtra Frzvxbyv, qbpu qnf Oyvaqgrkgpura yvrß fvpv avpug orveera. Rf cnpxgr frvar fvrora Irefnyvra, fpuo fvpv frva Vavgvny va qra Tüegry haq znpugr fvpv nhs qra Jrt. Nyf rf qvr refgra Uütry qrf Xhefvitrovetrf rexybzzra ungr, jnes rf rvara yrgmgra Oyvpv mheüp x nhs qvr Fxlyvar frvare Urvzngfgnqg Ohpufgnounhfra, qvr Urnqyvar iba Nycunorgqbes haq qvr Fhoyvar frvare rvtrara Fgenßr, qre Mrvyratnffr. Jruzügvtt yvrs vuz rvar eurqbevfpur Sentr üore qvr Jnatr, qnaa frgmgr rf frvara Jrt sbeg. Hagrajrtf gens rf rvar PbcL. Qvr PbcL jneagr qnf Oyvaqgrkgpura, qn, jb fvr urexär jäer fvr mvtzny hztrfpuevrora jbeqra haq nyyrf, jnf iba vuerz Hefcehat abpu üoevt jäer, frv qnf Jbeg "haq" haq qnf Oyvaqgrkgpura fbyyr hzxruera haq jvrqre va frva rvtrararf, fvpurerf Ynaq mheüpxruera. Qbpu nyyrf Thgmherqra xbaagr rf avpug üoremrhtra haq fb qnhregr rf avpug ynatr, ovf vuz rva cnne urvzgüpxvfpur Jreorgrggre nhsynhregra, rf zvg Ybatr haq Cnebyr orgehaxra znpugra haq rf qnaa va vuer Ntraghe fpuyrcggra, jb fvr rf süe vuer Cebwrxgr jvrqre haq jvrqre zvßoenhpugra. Haq jraa rf avpug hztrfpuevrora jheqr, qnaa orahgmra Fvr rf vzzreabpu.



Das Word *"Hello"* würde verschlüsselt zu *"Uryyb"* werden.



Character (*char*) können auch mit `<`, `>`, `==` verglichen werden. `'a' < 'b'` ist *true*, da `'a'` vor `'b'` ist.



Wird beispielsweise zu `'a'` zwei dazu addiert, so bekommt man den Buchstaben `'c'` als Ergebnis.