Name: Sahib Niyazov, Matrikelnummer: 217205805
Name: Tobias Baumann, Matrikelnummer: 215204750
Name: Karim Elrgl, Matrikelnummer: 219204060

Aufgabe 1
Für Variablennamen gelten einige Regeln. Beginnen müssen sie entweder mit einem Buchstaben oder einem Unterstrich (bspw. _halllo oder hallo_). Daran kann sich eine beliebige Abfolge von Buchstaben, Zahlen (0-9) oder Unterstrichen anreihen. Es wird ebenfalls zwischen Groß- und Kleinschreibung unterschieden (bspw. Ax ist nicht dasselbe wie aX). Leerzeichen, Interpunktionszeichen oder andere Sonderzeichen sind hingegen nicht erlaubt. (bspw. ~ ` ! @ # $ % ^ & * ( ) ; – : " ' < > , . ? / { } [ ] + = /). Daher ist der Name foobar erlaubt, während der Name foo&bar nicht akzeptiert werden kann.

Aufgabe 2
Namen, die nur programmintern auftreten, haben in den häufigsten Fällen 31 signifikante Stellen (ANSI-Standard). Sie können aus den Buchstaben A-Z und a-z, sowie aus den Ziffern 0-9 bestehen. Bei Namen, die als Querbezug zu Programmen in anderen Quelldateien dienen, sollte man sich allerdings auf maximal 6 Zeichen beschränken. Schlüsselwörter wie else, if, float, int etc. sind hingegen reserviert und müssen kleingeschrieben werden.

Aufgabe 5.1

```
#include <stdio.h>

main() {
int w, x, y, z; // w,x y,z wurden als Integer definiert
printf("W? ");  // Ausgabe W
scanf("%d",&w); // Eingabe W z.B 3
printf("X? ");  // Ausgabe X
scanf("%d",&x); // Eingabe X z.B 2
y=1;            // y hat die Werte 1

while(y<=x) y=y*w; // Schleife läuft so lange, bis x>=y ist, y muss neue Werte durch y*w
bekommen. In unserem Fall bekommt y den neuen Wert 3 und das Programm geht aus der
Schleife raus.
y=y/w;          // y bekommt hier wieder den Wert 1, Programm geht in eine neue Schleife.
while(y > 0) {  // in diesem Fall ist y gleich 1 und deswegen bleibt das Programm in der
Schleife.
z = x/y;        // z bekommt den Wert 2.
printf("%d",z); // Ausgabe Z. In unserem Fall wird 2 gezeigt.
x = x - y*z;    // x bekommt neuen Wert, laut unseres Beispiels den Wert 0.
y = y/w;        // y bekommt neuen Wert, laut unseres Beispiels 1/3, sodass 0 angezeigt wird,
weil y integer ist.
}               // dadurch, dass y=0 ist, geht das Programm aus der Schleife raus.
printf("\n");   // Ausgabe Leerzeichen.
}               // Ende des Programms. Als Ergebnis wird 2 angezeigt. (zwei & Leerzeichen
erscheint als Ergebnis).
```