

Imperative Programmierung  
Aufgabenblatt A02

1)

Eine Variable in C muss eine Zeichenkette des Typen Char sein. Es dürfen keine Sonderzeichen oder Zeichen verwendet werden, die für Operationen reserviert sind. Deshalb kann "foobar" ein Variablenname sein. "foo&bar" kann kein Variablenname sein, weil "&" für die Operation Modulus reserviert (und ein Sonderzeichen) ist.

2)

Es kann bei sehr langen Variablennamen zu Problemen kommen, weil diese vom Typ Char sind und dieser Typ nur einen Wertebereich von -128 bis 127 hat.

5)

a)

Das Programm rechnet eine Dezimalzahl in eine Zahl zu einer selbstbestimmten Basis um. In der Variablen "w" wird die gewünschte Basis geschrieben. Dies passiert durch eine Eingabe des Anwenders. In der Variablen "x" wird die umzuwandelnde Dezimalzahl gespeichert. Auch dies passiert durch eine Eingabe des Anwenders. Aus den beiden Eingaben wandelt das Programm nun die Dezimalzahl in die Zahl zur Basis "w" um.

Beispiel:

Man gebe für "w" die Zahl 2 und für "x" die Zahl 8 ein. So erhält man 1000.

b)

Anfangs deklariert das Programm die Variablen "w", "x", "y" und "z" als int Werte. Daraufhin werden die Variablen "w" und "x" vom Anwender definiert und "y" auf 1 gesetzt. Es beginnt eine Schleife, die läuft solange "y<=x", in der "y" mit "x" multipliziert und in "y" übergeben wird bis "y" größer "x" ist. Ist dies abgeschlossen, wird "y" durch "w" geteilt und in "y" übergeben. Es beginnt eine neue Schleife solange "y" größer 0 ist in welcher "x" durch "y" geteilt und in "z" übergeben und ausgegeben wird und es wird "x-y\*z" gerechnet und in "x" übergeben sowie "y/w" welches das Ergebnis in "y" übergibt. Diese Schleife wird solange ausgeführt bis "y" kleiner 0 ist.

Beispiel:

Man übergebe für "w" die Zahl 2 und für "x" die Zahl 8. Es beginnt die erste Schleife und da  $8(x) > 1(y)$ , wird zuerst "w" mit 1 multipliziert und mit "y" überschrieben bis das Ergebnis 16 (y) und somit  $16 \leq 8(x)$  ist und anschließend wird 16 (y) durch 2 (w) gerechnet und das Ergebnis 8 wird in "y" gespeichert.

Anschließend beginnt die nächste Schleife in welcher "y" größer 0 sein muss. Es wird  $8(x)$  durch 8 (y) geteilt und das Ergebnis 1 wird in "z" gespeichert und ausgegeben. Daraufhin wird  $8(x) - 8(y) * 1(z)$  gerechnet und 0 in "x" gespeichert. Anschließend wird 8 (y) durch 2 (w) geteilt und 4 in "y" gespeichert.

Anschließend beginnt die Schleife von Neuem, da  $4(y) > 0$  ist. Es wird  $0(x)$  durch 4 (y) geteilt und 0 in "z" gespeichert und ausgegeben.  $0(x) - 4(y) * 0(z)$  ergibt 0 und wird in "x" gespeichert. Danach wird 4 (y) durch 2 (w) geteilt und das Ergebnis 2 in "y" übergeben. Die Schleife fängt erneut an, da  $2(y) > 0$  ist. Es wird  $0(x)$  durch 2 (y) geteilt und 0 in "z" gespeichert und ausgegeben.  $0(x) - 2(y) * 0(z)$  ergibt 0 und wird in "x" gespeichert. Daraufhin wird 2 (y) durch 2 (w) geteilt und das Ergebnis 1 in "y" gespeichert.

Es beginnt die Schleife ein letztes Mal, da  $1(y) > 0$  gilt. Es wird  $0(x)$  durch 1 (y) geteilt und 0 in "z" gespeichert und ausgegeben.  $0(x) - 1(y) * 0(z)$  ergibt 0 und wird in "x" gespeichert.

Anschließend wird 1 (y) durch 2 (w) geteilt und das Ergebnis 0 (weil Typ int) in "y" gespeichert.

Nun ist die Bedingung für die while-Schleife nicht mehr gegeben, da  $0(y) > 0$  nicht erfüllt ist. Somit lautet das Endergebnis 1000.