Fachhochschule
Dortmund
University of Applied Sciences and Arts

Prof. Dr. Dirk Wiesmann
Programmierkurs 2
Praktikum 1

1. Als Aufwärmübung bearbeiten Sie bitte die folgenden Teilaufgaben.

   a) Starten Sie auf dem Praktikumsrechner den VNC-Viewer und stellen Sie eine Verbindung zum Linux-Server (LinuxPR) 172.22.1.40:5942 her. Erstellen Sie einen neuen Ordner und speichern Sie dort die Dateien `ggt.c`, `ggt.h` und `calcggt.c` aus der Vorlesung.

   b) Erzeugen Sie nun mit dem Compiler `gcc` die Objektdatei `ggt.o`. Danach erzeugen Sie das ausführbare Programm `calcggt` und starten das Programm.

   c) Erzeugen Sie ein `makefile`, welches die bedarfsgerechte Compilierung des Programms `calcggt` ermöglicht. Erzeugen Sie dann mit `make` eine Programmversion. Modifizieren Sie nacheinander die einzelnen Quellcode-Dateien, indem Sie z.B. einen Kommentar der Form `/* Test */` einfügen. Starten Sie dann `make` erneut. Welche Programmteile werden neu übersetzt?

2. Für die folgenden Aufgaben erstellen Sie bitte jeweils eine eigene Datei für die Definition der Funktionen. Erzeugen Sie zudem zugehörige Header-Dateien für Funktionsdeklarationen. Fassen Sie die Objektdateien der einzelnen Aufgaben zu einer *Library* zusammen. Dazu verwenden Sie das Archivprogramm `ar`. Eine *Library* `libmy.a` erzeugen Sie nach dem Muster

   ```
   ar r libmy.a obj1.o obj2.o
   ```

   Schreiben Sie zusätzlich eine Datei `main.c`, welche das Hauptprogramm enthält und jede der programmierten Funktionen mindestens einmal aufruft. Mit der Compileroption `-Llib` wird im Verzeichnis `lib` nach *Library*-Dateien gesucht. Mit der Option `-lmy` wird die *Library* `libmy.a` eingebunden. Das Präfix `lib` und die Dateiendung `.a` werden automatisch ergänzt. Für das Binden des Hauptprogramms verwenden Sie die erzeugte *Library*. Erstellen Sie ein geeignetes `makefile`.

3. Schreiben Sie eine Funktion, die jeden ASCII-Code zwischen 32 und 126 zusammen mit dem zugehörigen Zeichen auf dem Bildschirm ausgibt. Über einen Parameter s kann bestimmt werden, dass die Ausgabe in s Spalten erfolgt. Hier eine Beispielausgabe mit $s = 10$.

```
 32    33 !  34 "  35 #  36 $  37 %  38 &  39 ’  40 (  41 )
 42 *  43 +  44 ,  45 -  46 .  47 /  48 0  49 1  50 2  51 3
 52 4  53 5  54 6  55 7  56 8  57 9  58 :  59 ;  60 <  61 =
 62 >  63 ?  64 @  65 A  66 B  67 C  68 D  69 E  70 F  71 G
 72 H  73 I  74 J  75 K  76 L  77 M  78 N  79 O  80 P  81 Q
 82 R  83 S  84 T  85 U  86 V  87 W  88 X  89 Y  90 Z  91 [
 92 \  93 ]  94 ^  95 _  96 ‘  97 a  98 b  99 c 100 d 101 e
102 f 103 g 104 h 105 i 106 j 107 k 108 l 109 m 110 n 111 o
112 p 113 q 114 r 115 s 116 t 117 u 118 v 119 w 120 x 121 y
122 z 123 { 124 | 125 } 126 ~
```

4. Schreiben Sie eine Funktion, die das Einmaleins auf der Konsole ausgibt. Die Funktion fragt zunächst das kleinste und das größte Einmaleins ab, das berechnet werden soll. Der Anwender muss die Werte über die Tastatur eingeben. Danach erfolgt eine formatierte Ausgabe. Hier ein Beispiel:

```
Einmaleins
Von:10
Bis:20
  10  11  12  13  14  15  16  17  18  19  20
  20  22  24  26  28  30  32  34  36  38  40
  30  33  36  39  42  45  48  51  54  57  60
  40  44  48  52  56  60  64  68  72  76  80
  50  55  60  65  70  75  80  85  90  95 100
  60  66  72  78  84  90  96 102 108 114 120
  70  77  84  91  98 105 112 119 126 133 140
  80  88  96 104 112 120 128 136 144 152 160
  90  99 108 117 126 135 144 153 162 171 180
 100 110 120 130 140 150 160 170 180 190 200
```

5. Schreiben Sie eine Funktion, welches eine ganze Zahl x als Parameter besitzt und die zugehörige Dualzahl mit 16 Bit auf dem Bildschirm ausgibt. Es sollen nur Parameter verarbeitet werden mit $0 \leq x \leq 65535$. Sie können die Funktion `pow(x,y)` aus der Mathematikbibliothek libm.a nutzen. Dazu müssen Sie die Header-Datei `math.h` einbinden und den Compiler mit der Option `-lm` zum Binden der Mathematikbibliothek anweisen.