Christian-Albrechts-Universität zu Kiel
Institut für Informatik

Multimedia Information Processing Group
Prof. Dr. Reinhard Koch
Dr.-Ing. Christoph Starke, M.Sc. Vasco Grossmann,
M.Sc. Sascha Clausen

Computersysteme

Wintersemester 2018/2019

Serie 3

Ausgabetermin: Freitag, 02.11.2018
Abgabetermin: Freitag, 16.11.2018, 08:00 Uhr im Schrein

Bitte klammern oder heften Sie Ihre Abgabeblätter geeignet zusammen und notieren Sie sowohl Ihre Namen als auch Ihre Gruppennummer auf der Abgabe!

Präsenzaufgaben

Aufgabe 1

(a) Bestimmen Sie die Gleitkommadarstellung der Zahlen $(15)_{10}$ und $(6,5)_{10}$ im IEEE 32-Bit-Format.

(b) Addieren Sie die beiden Gleitkommazahlen und geben Sie das Ergebnis als IEEE 32-Bit-Gleitkommazahl an.

(c) Multiplizieren Sie die beiden Gleitkommazahlen und geben Sie das Ergebnis als IEEE 32-Bit-Gleitkommazahl an.

Aufgabe 2

(a) Rechnen Sie von Hand die vierstellige Zahl $(1011)_2$ aus $B = 2$ nach $B' = 10$ mittels Hornerschema um. Schreiben Sie dabei auch explizit die notwendige Klammerdarstellung für die Umrechnung der vierstelligen Zahl $(b_3 b_2 b_1 b_0)$ hin.

(b) Gegeben ist der C-Code für das Horner-Schema, wobei die Binärzahl als `char`-String ziffernweise eingelesen wird. Erinnerung: Ein String wird immer mit einer „\0“ als Endekennung terminiert. Die Kommentare sind leider verloren gegangen. Interpretieren Sie den Code zeilenweise und beantworten Sie zudem folgende Fragen:

(i) Wie viele Stellen darf die eingegebene Zahl maximal haben?

(ii) Welche Zeilen umschließt das `while`-Statement und unter welcher Kondition bricht die `while`-Schleife ab?

(iii) Was bewirken die Zeilen 12 bis 25?

(iv) Wo findet die eigentliche Berechnung statt?

```c
#include <stdio.h>

int main()
{
  char binaerstring[100];                                   // Kommentar
  printf("Gib natuerliche Zahl mit Basis 2 ein: ");
  scanf("%s", binaerstring);                                // Kommentar
  int i = 0;                                                // Kommentar
  int ergebnis = 0;                                         // Kommentar
  while(binaerstring[i] != 0)                               // Kommentar
  {
    char b;                                                 // Kommentar
    if(binaerstring[i] == '1')                              // Kommentar
    {
      b = 1;
    }
    else if(binaerstring[i] == '0')                         // Kommentar
    {
      b = 0;
    }
    else                                                    // Kommentar
    {
      printf("Ungueltige Eingabe: \'%c\'\n", binaerstring[i]);
      return 1;                                             // Kommentar
    }
    ergebnis = ergebnis + b;                                // Kommentar

    i++;                                                    // Kommentar
    if(binaerstring[i] != 0)                                // Kommentar
    {
      ergebnis = ergebnis * 2;                              // Kommentar
    }
  }
  printf("Die Binaerzahl %s entspricht der Dezimalzahl %d.\n", binaerstring, ergebnis);
  return 0;
}
```

Hausaufgaben

Aufgabe 1

(a) Bestimmen Sie die Gleitkommadarstellung der Zahlen $(58)_{10}$ und $(11,15)_{10}$ im IEEE 32-Bit-Format.

(b) Addieren Sie die beiden Gleitkommazahlen und geben Sie das Ergebnis als IEEE 32-Bit-Gleitkommazahl an.

(c) Multiplizieren Sie die beiden Gleitkommazahlen und geben Sie das Ergebnis als IEEE 32-Bit-Gleitkommazahl an.

30 Punkte, je 10 Punkte

Aufgabe 2

Bestimmen Sie folgende rationale Gleitkommazahlen im IEEE 32-Bit-Format (sowohl ihre Repräsentation als auch ihren dezimalen Zahlwert, aus dem man die Genauigkeit zumindest ungefähr ersehen kann):

(a) -2^{-142};

(b) die größte positive Zahl.

20 Punkte, je 10 Punkte

Aufgabe 3

Schreiben Sie ein Programm, welches eine positive Zahl zur Basis 10 in eine beliebige Basis $B, B \in \{2,3,...,16\}$ umwandelt. Nutzen Sie die Divisionsmethode. Speichern Sie dabei jede berechnete Ziffer Z_i an der Stelle i in einem Feld vom Typ `char` ab, wobei Sie die maximale Stellenzahl mit 100 fest angeben können. Geben Sie anschließend die im Feld gespeicherten Ziffern in der korrekten Reihenfolge und der korrekten Stellenzahl aus. Für Schleifen verwenden Sie bitte die `while`-Schleife

Hinweis: die Ausgabe einer einzelnen Ziffer Z_i zur Basis B bis $B = 16$ kann ziffernweise über `printf()` mit Hexadezimalausgabe erfolgen.

(a) Berechnen Sie von Hand nach der Divisionsmethode die Konvertierung von 250 (Basis 10) nach Basis 4.

6 Punkte

(b) Programmieren Sie unter Vorlage der unten gezeigten Beispielausgabe eine Schnittstelle zur Eingabe der zu konvertierenden Zahl sowie der Basis jeweils als `integer` über `scanf`. Erinnerung:

`scanf("%d", &<variablenname>);  scanf("%s", <feldname>);`

Testen Sie, ob die Zahlen im Gültigkeitsbereich liegen und brechen ggf. mit `return 1;` ab.

14 Punkte

(c) Berechnen Sie die einzelnen Ziffern der Konvertierung und speichern Sie diese in dem Feld ab. Geben Sie die Ziffern der konvertierten Zahl über `printf()` aus. Geben Sie auch aus, wie viele Stellen die Zahl jetzt hat.

25 Punkte

(d) Testen Sie das Programm mit dem Beispiel aus (a) und weiteren Konvertierungen von 250 in die Basen $B = 2,5,7,13,16$. Erzeugen Sie einen (oder mehrere) Snapshot(s) ihrer Ausgaben.

5 Punkte

Schicken Sie den (die) Snapshot(s) sowie Ihr C-Programmfile zusätzlich zur schriftlichen (gedruckten) Abgabe im Schrein per E-Mail an Ihren Übungsgruppenleiter. Schreiben Sie in den Betreff bitte: „Computer Systeme S3-H-<Gruppennummer>-<Matrikelnummer1>-<Matrikelnummer2>.c“.

Beispielausgabe 1:

```
legolas{rk} gcc -o h3 S3-H.c
legolas{rk} ./h3
Gib natuerliche Zahl mit Basis 10 ein: 250
Konvertierung in welche Basis? Gueltige Basen: 2 bis 16: 10
Konvertiere die Zahl 250 aus Basis 10 nach Basis 10
Die Zahl ist 3-stellig und lautet: 250
```

Beispielausgabe 2:

```
legolas{rk} ./h3
Gib natuerliche Zahl mit Basis 10 ein: 250
Konvertierung in welche Basis? Gueltige Basen: 2 bis 16: 2
Konvertiere die Zahl 250 aus Basis 10 nach Basis 2
Die Zahl ist 8-stellig und lautet: 11111010
```

50 Punkte