Algorithmen und Datenstrukturen

6. Übungsblatt

Zeitraum: 24. – 28. November 2014

Übung 1 (AGS 4.1)
Gegeben sei das folgende *C*-Programm:

```
 1  #include <stdio.h>
 2
 3  void g(int *x, int y) {
 4    /* label1 */
 5    *x = *x + y; /* label2 */
 6  }
 7
 8  void f(int x, int *y) {
 9    int z; /* label3 */
10    if (x > 0) {
11      f(x - 1, &z); /* $2 */ /* label4 */
12      *y = x * z;
13    }
14    else *y = 1;
15    if (*y > 2) g(y, x); /* $3 */
16    /* label5 */
17  }
18
19  int main() {
20    int e, a;
21    scanf("%i", &e); /* label6 */
22    f(e, &a); /* $1 */ /* label7 */
23    printf("a = %d\n", a);
24    return 0;
25  }
```

(a) Geben Sie den Gültigkeitsbereich jedes Objektes des Programms an.
(b) Stellen Sie eine Rechnung des Programms für die Eingabe `e=2` als pulsierenden Speicher dar. Dokumentieren Sie hierzu bei jedem Passieren eines Haltepunkts `labelX` jeweils alle *sichtbaren* Variablen mit ihren Wertebelegungen. Nutzen Sie die bereits im Programm eingetragenen Rücksprungmarken.

Übung 2 (AGS 3.1.4)
Für $n, k \in \mathbb{N}$ mit $k \leq n$ ist der Binomialkoeffizient $b(n,k)$ definiert durch:

$$b(n,k) = \binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Schreiben Sie ein C-Programm, das Binomialkoeffizienten berechnet. Überlegen Sie sich Problemlösungen, die es erlauben, dass Ihr C-Programm möglichst große Zahleneingaben korrekt verarbeiten kann.

Übung 3 (AGS 3.1.13)
Gegeben sei folgender Rumpf eines C-Programms:

```
1  #include <stdio.h>
2
3  // maximale Laenge der Zeichenketten
4  #define MAXLAENGE 31
5  // maximale Anzahl von Kontakten
6  #define MAXKONTAKTE 20
7
8  /*
9   * Gibt die einzelnen Komponenten des uebergebenen Personenfeldes aus.
10  */
11 void ausgabeAufKonsole(/* TODO */) {
12   for (i = 0; i < MAXKONTAKTE; i++) {
13     /* TODO Ausgabe der Kontakte auf Konsole */
14   }
15 }
16
17 /*
18  * Schreibt die Inhalte des Personenfeldes in die Datei personen.csv.
19  */
20 void schreibeInDatei(/* TODO */) {
21   FILE *datei = fopen("personen.csv", "w");
22
23   if (datei == NULL) {
24     printf("Fehler beim Oeffnen der Datei.");
25     return;
26   }
27
28   fprintf(datei, "Vorname;Name;PLZ;Ort;E-Mail;\n");
29
30   for (i = 0; i < MAXKONTAKTE; i++) {
31     /* TODO Ausgabe der Kontakte in Datei */
32   }
33
34   fclose(datei);
35 }
36
37 int main() {
38   for (i = 0; i < MAXKONTAKTE; i++) {
39     /* TODO Eingabe der Kontakte von Konsole */
40   }
41 }
```

(a) Überlegen Sie sich einen Strukturdatentyp in C, der Personen mit Vor- und Nachnamen, Postleitzahl, Wohnort sowie E-Mail-Adresse speichern kann.

(b) Nutzen Sie diesen Datentyp, um im obigen Programm die Ein- und Ausgabe von Personen zu realisieren. Vervollständigen Sie dazu die Funktionen `main` und `ausgabeAufKonsole`.

(c) Ergänzen Sie nun auch die Funktion `schreibeInDatei`. Schreiben Sie dazu pro Zeile die Komponenten einer Person durch Semikola getrennt in eine Datei. Öffnen Sie testweise die Datei `personen.csv` mit einem Tabellenkalkulationsprogramm.
Hinweis: Verwenden Sie den Aufruf `fprintf(datei, <args>)` statt `printf(<args>)` (vergleiche Zeile 28).

Übung 4 (AGS 4.4)
Gegeben sei das folgende C-Programm:

```
1  #include <stdio.h>
2  int c;
3
4  void f(int x, int y, int *z) {
5    /* label3 */
6    if (x > 5) {
7      f(x - 2, y, z); /* $2 */
8      *z = *z * x; /* label4 */
9    }
10   else *z = y; /* label5 */
11 }
12
13 int main() {
14   int a, b;
15   printf("\n Zahl a: ");
16   scanf("%d", &a);
17   printf("\n Zahl b: ");
18   scanf("%d", &b); /* label1 */
19   f(a, b, &c); /* $1 */ /* label2 */
20   printf("\n Das Ergebnis lautet: %d\n\n", c);
21   return 0;
22 }
```

(a) Geben Sie den Gültigkeitsbereich jedes Objektes des Programms an.
(b) Stellen Sie eine Rechnung des Programms für die Eingabe `a=8` und `b=1` als pulsierenden Speicher dar. Dokumentieren Sie hierzu bei jedem Passieren eines Haltepunkts `labelX` jeweils alle *sichtbaren* Variablen mit ihren Wertebelegungen. Nutzen Sie die bereits im Programm eingetragenen Rücksprungmarken.

Zusatzaufgabe 1 (AGS 4.5)
Gegeben sei das folgende C-Programm:

```
1  #include <stdio.h>
2
3  void f(int x, int y, int *z) {
4    /* label1 */
5    if (x > 0) {
6      f(x - 1, y, z); /* $2 */ /* label2 */
7      *z = x + y;
8    }
9    else *z = y;
10   /* label3 */
11 }
12
13 int main() {
14   int a, b, c;
15   printf("\n Zahl a: ");
16   scanf("%d", &a);
17   printf("\n Zahl b: ");
18   scanf("%d", &b); /* label4 */
19   f(a, b, &c); /* $1 */ /* label5 */
20   printf("\nDas Ergebnis lautet: %d\n\n", c);
21   return 0;
22 }
```

(a) Geben Sie den Gültigkeitsbereich jedes Objektes des Programms an.

(b) Stellen Sie eine Rechnung des Programms für die Eingabe `a=2` und `b=3` als pulsierenden Speicher dar. Dokumentieren Sie hierzu bei jedem Passieren eines Haltepunkts `labelX` jeweils alle *sichtbaren* Variablen mit ihren Wertebelegungen. Nutzen Sie die bereits im Programm eingetragenen Rücksprungmarken.