

Angewandte Informatik – Klausur – 29. Januar 2016

Prof. Dr. Peter Gerwinski, Wintersemester 2015/16

Name:	
Matrikel-Nr.:	

Zeit: 120 Minuten

Zulässige Hilfsmittel:

- Schreibgerät
- Beliebige Unterlagen in Papierform und/oder auf Datenträgern
- Elektronische Rechner (Notebook, Taschenrechner o. ä.)
ohne Zugang zu Datennetzen jeglicher Art

Nur die o. a. zulässigen Hilfsmittel dürfen sich während der Klausur im Arbeitsbereich befinden. WLAN-, Bluetooth- und sonstige Funkeinheiten von Notebooks o. ä. sind auszuschalten; ggf. dafür vorhandene physische Schalter sind zu benutzen. Mobiltelefone, Geräte mit mobilem Internet-Zugang u. ä. sind auszuschalten und in der Tasche zu verstauen.

Die reguläre Maximalpunktzahl beträgt 42 Punkte.

Bei besonderen Leistungen sind Zusatzpunkte möglich.

Mit 20 erreichten Punkten gilt die Klausur als bestanden.

Die Beispielprogramme werden Ihnen auf Datenträger (USB-Stick) zur Verfügung gestellt. Die Abgabe von digital gelösten Aufgaben hat auf demselben Datenträger zu erfolgen.

Aufgabe 1: Strings

Wir betrachten das folgende Programm (aufgabe-1.c):

```

1  #include <stdio.h>
2  #include <string.h>
3
4  void fun (char *s)
5  {
6      int len = strlen (s);
7      for (int i = 0; i < len; i++)
8          for (int j = i + 1; j < len; j++)
9              if (s[i] > s[j])
10                 {
11                     char sx = s[i];
12                     s[i] = s[j];
13                     s[j] = sx;
14                 }
15  }
16
17  int main (void)
18  {
19      char s[] = "Informatik";
20      fun (s);
21      printf ("%s\n", s);
22      return 0;
23  }

```

Auf einem Rechner, der den ASCII-Zeichensatz verwendet, lautet die Ausgabe: `Iafikmnort`

- (a) Was bewirkt die Funktion `fun`, und wie funktioniert sie? (4 Punkte)

- (b) Warum beginnt der Ausgabe-String mit 'l' und nicht mit 'a'? (2 Punkte)

- (c) Was passiert, wenn die Zeile `char s[] = "Informatik";` durch `char s[] = { 'l', 'n', 'f', 'o' };` ersetzt wird, und warum? (2 Punkte)

- (d) Von welcher Ordnung (Landau-Symbol) ist die Funktion fun und warum?

Wir beziehen uns hierbei auf die Anzahl der Vergleiche `s[i] > s[j]` in Abhängigkeit von der Länge des Eingabe-Strings `"Informatik"`. (1 Punkt)

- (e) Beschreiben Sie – in Worten und/oder als C-Quelltext –, wie sich die Funktion `fun` hinsichtlich der Rechenzeit effizienter gestalten läßt. Von welcher Ordnung (Landau-Symbol) ist Ihre effizientere Version der Funktion und warum? (4 Punkte)

Aufgabe 2: Zeigerarithmetik

Wir betrachten das folgende Programm (aufgabe-2.c):

```

1  #include <stdio.h>
2  #include <stdint.h>
3
4  void output (uint16_t *a)
5  {
6      for (int i = 0; a[i]; i++)
7          printf ("_%d", a[i]);
8      printf ("\n");
9  }
10
11 int main (void)
12 {
13     uint16_t prime_numbers[] = { 2, 3, 5, 7, 11, 13, 17, 0 };
14
15     uint16_t *p1 = prime_numbers;
16     output (p1);
17     p1++;
18     output (p1);
19
20     char *p2 = prime_numbers;
21     output (p2);
22     p2++;
23     output (p2);
24
25     return 0;
26 }

```

Das Programm wird kompiliert und ausgeführt:

```
$ gcc -Wall -std=c99 aufgabe-2.c -o aufgabe-2
aufgabe-2.c: In function 'main':
aufgabe-2.c:20:13: warning: initialization from
                    incompatible pointer type [enabled by default]
aufgabe-2.c:21:3: warning: passing argument 1 of 'output' from
                    incompatible pointer type [enabled by default]
aufgabe-2.c:4:6: note: expected 'uint16_t *' but argument is of type 'char *'
aufgabe-2.c:23:3: warning: passing argument 1 of 'output' from
                    incompatible pointer type [enabled by default]
aufgabe-2.c:4:6: note: expected 'uint16_t *' but argument is of type 'char *'
$ ./aufgabe-2
 2 3 5 7 11 13 17
 3 5 7 11 13 17
 2 3 5 7 11 13 17
768 1280 1792 2816 3328 4352
```

(a) Erklären Sie die Funktionsweise der Funktion `output()`. (2 Punkte)

Aufgabe 3: Objektorientierte Programmierung

Wir betrachten das folgende Fragment ([aufgabe-3.c](#)) eines objektorientierten Mathematik-Programms:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct
{
    char symbol;
    int (*calculate) (int a, int b);
}
operation;

operation *new_operation (void)
{
    operation *op = malloc (sizeof (operation));
    op->symbol = '?';
    op->calculate = NULL;
    return op;
}

[...]

int main (void)
{
    operation *op[4];
    op[0] = new_plus ();
    op[1] = new_minus ();
    op[2] = new_times ();
    op[3] = NULL;
    for (int i = 0; op[i]; i++)
        printf ("2_ %c_3_ = _%d\n", op[i]->symbol, op[i]->calculate (2, 3));
    return 0;
}
```

- (a) Was bedeutet `int (*calculate) (int a, int b)`, und für welchen Zweck wird es verwendet? (3 Punkte)

- (b) Ergänzen Sie das Fragment zu einem funktionsfähigen Programm, das die folgende Ausgabe erzeugt:

$$\begin{array}{l} 2 + 3 = 5 \\ 2 - 3 = -1 \\ 2 * 3 = 6 \end{array}$$

(5 Punkte)

Abgabe auf Datenträger ist erwünscht, aber nicht zwingend.

Für Notizen verwenden Sie nötigenfalls die Rückseiten der Klausurbögen und/oder zusätzliche Blätter.

(c) Was geschieht, wenn man auf die Zeile `op[3] = NULL;` verzichtet, und warum? (2 Punkte)



Aufgabe 4: XBM-Grafik

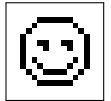
Bei einer XBM-Grafikdatei handelt es sich um ein als C-Quelltext abgespeichertes Array, das die Bildinformationen enthält:

- Jedes Bit entspricht einem Pixel.
- Nullen stehen für Weiß, Einsen für Schwarz.
- LSB first.
- Jede Zeile des Bildes wird auf ganze Bytes aufgefüllt.
- Breite und Höhe des Bildes sind als Konstantendefinitionen (**#define**) in der Datei enthalten.

Sie können eine XBM-Datei sowohl mit einem Texteditor als auch mit vielen Grafikprogrammen öffnen und bearbeiten.

Beispiel ([aufgabe-4.xbm](#)):

```
#define aufgabe_4_width 14
#define aufgabe_4_height 14
static unsigned char aufgabe_4_bits[] = {
    0x00, 0x00, 0xf0, 0x03, 0x08, 0x04, 0x04, 0x08, 0x02, 0x10, 0x32, 0x13,
    0x22, 0x12, 0x02, 0x10, 0x0a, 0x14, 0x12, 0x12, 0xe4, 0x09, 0x08, 0x04,
    0xf0, 0x03, 0x00, 0x00 };
```



Ein C-Programm, das eine XBM-Grafik nutzen will, kann die `.xbm`-Datei mit **#include "..."** direkt einbinden.

Schreiben Sie ein Programm, das die XBM-Datei als ASCII-Grafik ausgibt, z. B.:

```

  * * * * *
 *           *
*           *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
  * * * * *
```

(8 Punkte)

Abgabe auf Datenträger ist erwünscht, aber nicht zwingend.

Für Notizen verwenden Sie nötigenfalls die Rückseiten der Klausurbögen und/oder zusätzliche Blätter.

Viel Erfolg!