Dipl.Phys. Gerald Kempfer

Berliner Hochschule für Technik
- University of Applied Sciences -
Fachbereich VI – Informatik und Medien
Studiengang Technische Informatik Bachelor

Beispiel-Klausur zur Lehrveranstaltung Grundlagen der Informatik (TI-B) Wintersemester 2023/2024

Hinweise:

Die Teilnahme an der Klausur ist nur bei Bestehen der Übungsaufgaben zulässig!

Zum Bestehen der Klausur benötigen Sie 50 Punkte von 100 möglichen Punkten.

Die Bearbeitungszeit der Klausur beginnt pünktlich um 08:00 Uhr und endet – ebenfalls pünktlich – um 09:30 Uhr. Sie haben also 90 Minuten Zeit.

Folgende Hilfsmittel in Papierform sind zugelassen: Skripte, Mitschriften und Bücher. **Nicht** zugelassen sind elektronische Geräte (Handys, Notebooks, PDAs, usw.).

Schreiben Sie mit Kugelschreiben oder Füller, **nicht** mit Bleistift! Verwenden Sie **nicht** die Farbe Rot. Schreiben Sie leserlich – was ich nicht lesen kann, ist grundsätzlich falsch! Beschreiben Sie nur die Vorderseiten!

Jeder Austausch mit anderen Personen wird als Täuschungsversuch gewertet und führt dazu, dass die Klausuren aller Beteiligten als „nicht bestanden“ gewertet werden.

Erläuterungen sollten kurz, aber dennoch präzise und vollständig sein. Wenn möglich ist eine stichpunktartige Beantwortung zu wählen, sofern die Verständlichkeit gegeben ist. Im Zweifelsfall können ganze Sätze Klarheit schaffen.

Kennzeichnen Sie jedes Blatt, das Sie abgeben, links oben mit Ihrer Matrikelnummer.

Viel Erfolg!

Name: _____

Matrikelnummer: _____ letzter Prüfungsversuch: ☐

Bewertung:

Aufgabe	Mögliche Punktzahl	Erreichte Punktzahl
1	10	
2	15	
3	20	
4	25	
5	30	
Summe	**100**	

Note Klausur _____

Aufgabe 1: Finden Sie im folgenden Programm alle Syntaxfehler. (___ / 10)
Markieren Sie die Fehler und schreiben Sie hinter bzw. unter die Zeile, wie die Zeile richtig aussehen muss.

```
01   #include <stdio.h]
02
03   void Print(char *);
04   int GetMax(int, int)
05
06   int main();
07   {
08      int a = 0, b = 15;
09      char *Fehler = 'Dies ist falsch!";
10
11      for (a < b; a++)
12      {
13         printf("%s , Fehler);
14         printf("%f\n", GetMax(a, b);
15      }
16      print("Fertig! /* Programmende */ ");
17      return 0;
18   }
19
20   int Print(char *Text)
21   {
22      return printf(Text);
23   }
24
25   int GetMax(int i, int j)
26   {
27      return (i > j) ? j; i;
28   }
```

Aufgabe 2: Multiple-Choice-Fragen. (___ / 15)
Kreuzen Sie an, wie der Satz richtig heißen muss. Es gibt immer nur eine richtige Antwort.

Folgenden Datentypen gibt es nicht:
- ❑ `short double`
- ❑ `long double`
- ❑ `unsigned short int`

Der Ausdruck `(1 == 3) + (5 == 5)` ergibt
- ❑ 0.
- ❑ 1.
- ❑ einen Compilerfehler.

Vor dem C99-Standard wurde ein Wahrheitswert üblicherweise gespeichert in einer Variablen vom Datentyp
- ❑ `float`
- ❑ `double`
- ❑ `int`

Um ein 3x3 Array von `char` zu definieren, wird folgende Zeile verwendet:
- ❑ `char Array[3; 3];`
- ❑ `char Array[3][3];`
- ❑ `char Array[3, 3];`

`void` ist ein
- ❑ Datentyp.
- ❑ konstanter Wert.
- ❑ Operator.

Welchen Begriff gibt es nicht?
- ❑ Funktionsdeklaration
- ❑ Funktionsinitialisierung
- ❑ Funktionsdefinition

Mit dem binären Plusoperator werden Werte addiert. Nicht addiert werden können:
- ❑ Zahlen
- ❑ Zeichen
- ❑ Zeichenketten

Wie wird die dezimale Zahl 42 hexadezimal dargestellt?
- ❑ 0x29
- ❑ 0x2a
- ❑ 0x2b

Eine Initialisierung ist das Setzen einer Variable auf einen Startwert bei
- ❑ der Definition.
- ❑ der Deklaration.
- ❑ der Zuweisung.

Ein Byte besteht aus
- ❑ 1 Bit.
- ❑ 8 Bit.
- ❑ 16 Bit.

Der Operator für ein "logisches Und" lautet
- ❑ `AND`
- ❑ `||`
- ❑ `&&`

Welche Kontrollstruktur ist keine Schleife?
- ❑ `for`
- ❑ `switch`
- ❑ `while`

Welcher Zahlenwert entspricht dem Wahrheitswert falsch?
- ❑ 0
- ❑ jeder Zahlenwert ungleich 0
- ❑ jeder Zahlenwert ungleich 1

Welches der folgenden Schlüsselwörter beendet eine `switch`-Anweisung?
- ❑ `case`
- ❑ `return`
- ❑ `default`

Welche der folgenden Ausdrücke liefert keinen Compilerfehler?
- ❑ `printf('Fehler!');`
- ❑ `printf("Fehler!"):`
- ❑ `printf("Fehler!");`

Aufgabe 3: Was gibt das folgende Programm aus? (___ / 20)
Schreiben Sie die Ausgaben auf das folgende Blatt!

```
#include <stdio.h>

int  F1( char , int * );
int  F2( char );
char F3( int );

int main()
{
   int a = -1;

   printf("%cei", F3(F1('A', &a)));
   printf("%cp",  F3(F3('q')));
   printf("%ce",  F3(F1('g', &a)));
   printf("%ck",  F1(F3('i'), &a));
   printf("%ca",  'k' + F3(F2('B')));
   printf("%cs",  F3(F1('q', &a)));
   printf("%cr",  F3(F1('p', &a)));

   return 0;
}

int F1(char Zeichen, int *Zahl)
{
   *Zahl += 1;
   return (Zeichen + *Zahl);
}

int F2(char Zeichen)
{
   --Zeichen;
   return (Zeichen - 'A');
}

char F3(int Zahl)
{
   return (Zahl += 1);
}
```

Zwischenergebnisse:

(Geben Sie die Funktionsergebnisse folgender Aufrufe an!) (15 Punkte)

```
F1('A', &a)                    = _____
F3(F1('A', &a))                = _____

F3('q')                        = _____
F3(F3('q'))                    = _____

F1('g', &a)                    = _____
F3(F1('g', &a))                = _____

F3('i')                        = _____
F1(F3('i'), &a)                = _____

F2('B')                        = _____
F3(F2('B'))                    = _____
'k' + F3(F2('B'))              = _____

F1('q', &a)                    = _____
F3(F1('q', &a))                = _____

F1('p', &a)                    = _____
F3(F1('p', &a))                = _____
```

Bildschirm-Ausgabe des oben stehenden Programms:

(Beachten Sie dabei auch die Formatierungsangaben!) (5 Punkte)

Aufgabe 4: Schreiben Sie die passende Funktion `ZaehleZiffern` zum vorgegebenen Hauptprogramm. Die Funktion erhält eine Zeichenkette. In dieser Zeichenkette sollen nur die Ziffern `0` ... `9` gezählt werden, alle anderen Zeichen sollen ignoriert werden. Die Anzahl der gezählten Ziffern soll als Funktionsergebnis zurückgegeben werden.
Verwenden Sie in der Funktion nur die Zeigerschreibweise!
Weitere Headerdateien sind nicht erlaubt! (___ / 25)

```
#include <stdio.h>

// Platz für Ihre Funktionsdeklaration (3 Punkte):




int main()
{
    char *Text = "Dieser 1 Text hat 5 Ziffern: 123!";

    printf("Im Text\n%s\n", Text);
    printf("sind %i Ziffern", ZaehleZiffern(Text));
    printf(" enthalten.\n");
}

// Platz für Ihre Funktionsdefinition (22 Punkte):
```

Aufgabe 5: Schreiben Sie die passenden Funktionen zum vorgegebenen Hauptprogramm.
Die erste Funktion `Trim` soll alle Leerzeichen am Ende einer Zeichenkette löschen (d.h. mit dem ASCII-Wert 0 überschreiben). Als Parameter wird die Zeichenkette übergeben. Die Funktion gibt nichts zurück.
Die zweite Funktion `countSpaces` soll alle Leerzeichen in einem Array von 10 Zeichenketten mit je maximal 50 Zeichen zählen. Als Parameter erhält es ein zweidimensionales Array von Zeichen. Das Funktionsergebnis ist die Anzahl der Leerzeichen in allen Zeichenkette des Arrays von Zeichenketten.
Weitere Headerdateien sind nicht erlaubt! (___ / 30)

```
#include <stdio.h>

// Platz für Ihre Funktionsdeklarationen (6 Punkte):




int main()
{
   char Text[10][50] = {"Dies ist ein langer      ",
                        "Text, der in mehreren    ",
                        "Zeilen untergebracht     ",
                        "ist und der auch noch    ",
                        "in manchen Zeilen mehrere",
                        "Leerzeichen am Ende      ",
                        "beinhaltet! Die Leer-    ",
                        "zeichen dieses Textes    ",
                        "sollen gezählt werden!   ",
                        "                         "};
   int i = 0;

   for (i = 0; i < 10; i++)
      Trim(Text[i]);
   printf("\nIn dem Text sind %i Leerzeichen enthalten!\n",
           countSpaces(Text));
}

// Platz für Ihre Funktionsdefinitionen (24 Punkte):
```

Fortsetzung der Funktionsdefinitionen für Aufgabe 5: