

KLAUSUR im FACH *Ingenieur-Informatik*

im Wintersemester 2017/2018

Name, Vorname:

Studiengang/Semester:

Prüfer:

Dipl.-Ing. (FH) Marko Weber, Dipl.-Ing. (FH) Sebastian Balz, Dipl.-Ing. (FH) Rüdiger Ehret

Bearbeitungshinweise

1. Tragen Sie auf jeder Seite in der Kopfzeile Ihre Matrikelnummer ein.
2. Der Aufgabensatz (inkl. Deckblatt und Anhang), der aus 10 Seiten besteht (Seite 1 bis 12), ist auf Vollständigkeit zu überprüfen.
3. Der Aufgabensatz ist mit den Lösungsblättern abzugeben.
4. Lösungen auf selber mitgebrachten Lösungsblättern werden nicht ausgewertet. Verwenden Sie die Ihnen ausgeteilten Lösungsblätter und tragen Sie auch dort Ihre Matrikelnummer ein.
5. Bei Rechenaufgaben muss der Lösungsweg ersichtlich und lesbar sein, sonst erfolgt keine Bewertung der Aufgabe oder des Aufgabenteils.
6. Die Bearbeitungszeit beträgt 90 Minuten.
7. Es wird hiermit darauf hingewiesen, dass vom Prüfungsamt nicht vorher geprüft wurde, ob Sie das Recht bzw. die Pflicht zur Teilnahme an dieser Klausur haben. Die Teilnahme erfolgt auf eigene Gefahr, gleichzeitig bekundet die Teilnahme die Zustimmung zu diesem Passus.
8. Hilfsmittel:
 - handgeschriebene Formelsammlung (1 DIN A4 Blatt)
9. Bewertung:
 - Gesamtpunktzahl = 100 Punkte
 - Note 1,0 = 90 Punkte
 - Note 4,0 = 45 Punkte

Aufgabe	1	2	3	4	5	6	7	SUMME	
Punkte	10	15	10	10	20	15	20	100	NOTE
Erreichte Punkte									

Aufgabe 1: (10 Punkte)

Es ist jeweils eine Antwort richtig. Falsche Antworten führen **nicht** zu Punktabzug! Kreuzen Sie auf alle Fälle immer eine Antwort an!

1.1 Wie lautet die kleinste Informationseinheit?

- a) Byte
- b) **Bit**
- c) Nibble
- d) Crumb

1.2 Ein Megabyte (MB) entspricht

- a) 10^3 Byte
- b) **10^6 Byte**
- c) 10^9 Byte
- d) 10^{12} Byte

1.3 Ein _____ ergänzt verwendete Bibliotheken und setzt die einzelnen Objektdateien in ein ausführbares Programm um.

- a) Compiler
- b) Interpreter
- c) Präprozessor
- d) **Linker**
- e) Debugger

1.4 Welche Datei wird erzeugt, wenn man `cl /c main.c` aufruft?

- a) **main.obj**
- b) main.exe
- c) main.lib
- d) main.zip
- e) main.i

1.5 Was ist keine höhere Programmiersprache?

- a) **XML**
- b) C
- c) C++
- d) Java

1.6 Welche Dateierweiterung hat eine statische Bibliothek (z.B. die verwendete Utilities)?

- a) exe
- b) **lib**
- c) dll
- d) c
- e) sln

1.7 Was ist der Wertebereich, den man mit einer char Variablen darstellen kann?

- a) 0..256
- b) -127..128
- c) 0..255
- d) **-128..127**
- e) keiner der obigen Wertebereiche

1.8 Was darf bei einer Rekursion auf keinen Fall fehlen?

_____ **Abbruchkriterium**

1.9 Durch welches Schlüsselwort wird aus einer programmglobalen Variablen eine modulglobale Variable?

_____ **static**

1.10 Wie werden Arrays in C immer übergeben?

_____ **call by reference**

Aufgabe 2: (15 Punkte)

2.1 Wandeln Sie die folgenden positiven Zahlen (unsigned) in das jeweils andere Zahlensystem um: (**4 x 2 Punkte**)

$$\begin{array}{rcl}
 1100101101_2 & = & \underline{\hspace{2cm}}_8 \quad 1455 \\
 1100101101_2 & = & \underline{\hspace{2cm}}_{16} \quad 32D \\
 16_{18} & = & \underline{\hspace{2cm}}_{10} \quad 24 \\
 42_{10} & = & \underline{\hspace{2cm}}_2 \quad 101010
 \end{array}$$

2.2 Berechnen Sie das Zweierkomplement der unten gegebenen 8-Bit Zahl. Der Rechenweg muss ersichtlich sein! (**3 Punkte**)

Binärzahl:

0	1	0	0	1	1	1	0
MSB							LSB

Binärzahl: 01001110

Einerkomplement: 10110001

Zweierkomplement: 10110010

Zweierkomplement:

--	--	--	--	--	--	--	--

2.3 Führen Sie eine Subtraktion entsprechend den Regeln im Dezimalsystem durch. Der Rechenweg inkl. Überträge muss ersichtlich sein! (**4 Punkte**)

$$\begin{array}{r}
 11001000 \\
 - 10001101 \\
 \hline
 \end{array}$$

00111011

Aufgabe 3: (10 Punkte)

Gegeben sei folgende Struktur:

```
#define VALUES_PER_DAY 24
```

```
struct temp
{
    int temp_values[VALUES_PER_DAY]; // Stündl. Temperaturwerte eines Tages
    int max_temp;                    // Max. Temperatur des Tages
};
```

Die Struktur wird per „call by reference“ an die Funktion „calc_maxtemp“ übergeben. Die Funktion gibt den größten gemessenen Temperaturwert „max_temp“ eines Tages zurück. Zusätzlich soll der größte gemessene Temperaturwert auf dem Bildschirm ausgegeben werden. Vervollständigen Sie die Funktion „calc_maxtemp“ und stellen Sie sicher, dass die Variable „max_temp“ den größten gemessenen Temperaturwert enthält.

```
int calc_maxtemp( struct temp* tMonday )
{
    int i = 0;
    int maxTemp = 0;

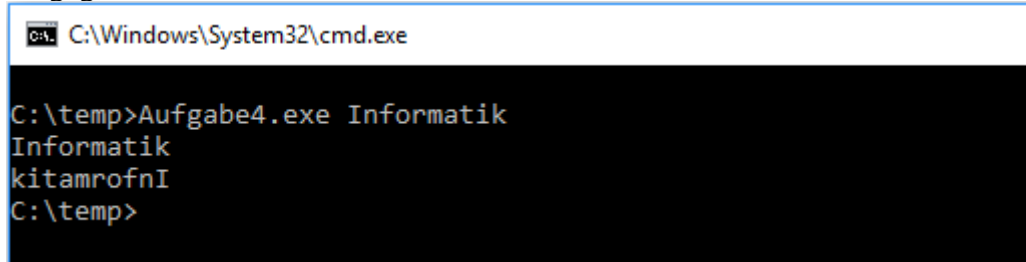
    if (tMonday)
    {
        tMonday->max_temp = tMonday->temp_values[0];
        for (i=1; i < VALUES_PER_DAY; i++)
        {
            if (tMonday->max_temp < tMonday->temp_values[i])
                tMonday->max_temp = tMonday->temp_values[i];
        }
        printf("%d", tMonday->max_temp);
        maxTemp = tMonday->max_temp;
    }

    return maxTemp;
}
```

```
}
```

Aufgabe 4: (10 Punkte)

Ein Programm soll von der Kommandozeile eine Zeichenkette übergeben bekommen (max 50 Zeichen) und diese auf dem Bildschirm ausgeben. Anschließend soll die übergebene Zeichenkette in umgekehrter Reihenfolge Zeichen für Zeichen (TIPP!) auf dem Bildschirm ausgegeben werden.



The screenshot shows a Windows command prompt window with the title bar 'C:\Windows\System32\cmd.exe'. The command prompt shows the following text: 'C:\temp>Aufgabe4.exe Informatik', followed by the output 'Informatik', 'kitamrofnI', and the prompt 'C:\temp>'.

Wird keine Zeichenkette übergeben, soll stattdessen die Ausgabe „Nicht genügend Parameter übergeben!“ auf dem Bildschirm erfolgen.

Vervollständigen Sie das untenstehende Programm:

```
void main ( int argc, char* argv[] )
{
    char zeichenkette[51];

    if (argc == 2)
    {
        strcpy( zeichenkette, argv[1] );

        printf("%s\n", zeichenkette);

        for (i = strlen(zeichenkette); i > 0; i--)
        {
            printf("%c", zeichenkette[i - 1]);
        }

    }
    else
    {
        printf("Nicht genügend Parameter übergeben!");
    }

    return EXIT_SUCCESS;
}
```

Aufgabe 5: (20 Punkte)

Schreiben Sie hinter die Kommentare, was *exakt* in der Konsole auf einem 32-Bit System ausgegeben wird. Im Kommentar sind die möglichen Punkte aufgeführt.

```
#include "MyVars.h"
```

```
void main(void)
{
    int k = 3;
    int i = 2;
    unsigned char uc1;
    unsigned char uc2;
    unsigned int parkingarray[MAX_STOCK_PER_ROW][10];
    char car_model[12] = "Passat";
    struct car myCar = {4711, VW, "Golf", "1223"};

    printf("%d\n", sizeof(k));           // 4 (1P)
    printf("%d\n", sizeof(i));           // 4 (1P)
    printf("%d\n", sizeof(uc1)+sizeof(uc1)); // 2 (1P)
    printf("%d\n", sizeof(parkingarray)); // 200 (1P)
    printf("%d\n", sizeof(PSTRUCTCAR));  // 4 (1P)
    printf("%d\n", sizeof(car_model));    // 12 (1P)
    printf("%s\n", car_model);            // Passat (1P)
    printf("%d\n", strlen(car_model));    // 6 (1P)
    printf("%d\n", MUL(k,i));             // 6 (1P)
    printf("%d\n", 2*MUL(k,i));           // 12 (1P)
    printf("%s\n", strtok(car_model, "s")); // Pa (2P)
    printf("%d\n", MUL(k++,++i));         // 9 (2P)
    printf("%d\n", sizeof(myCar));        // 44 (2P)
    printf("%d\n", k);                   // 4 (1P)

    uc1 = 0x09;
    uc2 = 0x0F;
    printf("%x\n", (unsigned char) (uc1 ^ (~uc2))); // F9 (3P)
}
```

```
#define MAX_STOCK_PER_ROW 5
#define MUL(a,b) a*b
enum brand {VW, Opel, Audi};
union config_number
{
    char standard[8];
    char variant[32];
};
struct car
{
    unsigned int car_id;
    enum brand name;
    char* model;
    union config_number serial;
};
struct bits
{
    unsigned char optid1 : 4;
    unsigned char optid2 : 5;
};
typedef struct car * PSTRUCTCAR;
```

MyVars.h

Aufgabe 6: (15 Punkte)

6.1 Gegeben sei die fehlerhafte C-Funktion „printTextFile“. Korrigieren Sie sämtliche Fehler innerhalb der Funktion. Die Funktion „outputText“ kann als gegeben betrachtet werden. (5 Punkte)

```
void printTextFile(char* filename)
{
    FILE* pf;
    pf = fopen("filename", "w");           // pf = fopen(filename, "r");

    if (pf == 0) // Error handling          // if (pf != 0)
    {
        outputText(pf);
    }
    else
    {
        printf("Fehler bei Öffnen der Datei: %d", filename ); // %s
    }

    fclose(FILE* pf); // Close the file    // fclose(pf);
}
```

6.2 Die Funktion „outputText“ soll den Inhalt der Textdatei auf dem Bildschirm ausgeben. Leere Zeilen sollen nicht auf dem Bildschirm ausgegeben werden. Implementieren Sie diese Funktion. (5 Punkte)

```
void outputText(FILE* pf)
{
    char buf[255];

    if (pf)
    {
        while (feof(pf) == NULL)
        {
            fgets(buf, 255, pf);
            if (buf[0] != '\n')
            {
                printf("%s", buf);
            }
        }
    }
}
```


6.3 Implementieren Sie die Suchfunktion „searchTextInFile“. Der Funktion wird ein Suchwort „searchString“ übergeben. Es sollen nur Zeilen aus einer Textdatei auf dem Bildschirm ausgegeben werden, welche das Suchwort enthalten. (5 Punkte)

TIPP: Funktion strstr() kann verwendet werden!

```
void searchTextInFile(FILE* pf, char* searchString)
{
    char buf[255];

    if (pf)
    {
        while (feof(pf) == NULL)
        {
            fgets(buf, 255, pf);
            if ((buf[0] != '\n') && strstr(buf, searchString))
            {
                printf("%s", buf);
            }
        }
    }
}
```

```
}
```

Aufgabe 7 (20 Punkte)

Die Quersumme einer Zahl soll sowohl iterativ als auch rekursiv berechnet werden.

Beispiel: $\text{Quersumme}(4711) = 4 + 7 + 1 + 1 = 13$

TIPP: Modulo Division verwenden!

7.1 Implementieren Sie eine iterative Lösung in die untenstehende Funktion (8 Punkte)

```
int digitsumIterative( int zahl )
```

```
{
```

```
    int result = 0;
```

```
    while (zahl > 0)
```

```
    {
```

```
        result += zahl % 10;
```

```
        zahl = zahl / 10;
```

```
    }
```

```
    return result; // Wert der Variablen entspricht Quersumme von „zahl“  
}
```

7.2 Implementieren Sie eine rekursive Lösung in die untenstehende Funktion (12 Punkte)

```
int digitsumRecursive( int zahl )
{
    int result = 0;

    if (zahl > 9)
        result = zahl % 10 + digitsumRecursive (zahl / 10);
    else
        result = zahl;
```

```
    return result; // Wert der Variablen entspricht Quersumme von „zahl“
}
```

Anhang: ASCII-Tabelle

Dez.	Hex.	Zeichen	Dez.	Hex.	Zeichen	Dez.	Hex.	Zeichen	Dez.	Hex.	Zeichen
0	0	NUL	32	20		64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(72	48	H	104	68	h
9	9	HT	41	29)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL