

KLAUSUR im FACH *Ingenieur-Informatik*

im Wintersemester 2015/2016

Name, Vorname:

Studiengang/Semester:

Prüfer:

Dipl.-Ing. (FH) Sebastian Balz, Dipl.-Ing. (FH) Sebastian Stelzl, 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 10), 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	10	10	20	15	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 Informatik ist die Wissenschaft von der systematischen Verarbeitung von ____ .

- a) Programmcode
- b) Dateien
- c) **Informationen**
- d) Datenstrukturen
- e) Diagrammen

1.2 C89 bedeutet, dass

- a) **es sich um den C-Standard von 1989 handelt**
- b) es sich um die C-Version mit den 89 Schlüsselworten handelt
- c) C 1989 erfunden wurde
- d) 89% von C standardisiert sind
- e) C basierend auf den frühen Arbeiten von Alain Turing (1889) beruht

1.3 Ein _____ übersetzt zeilenweise den Programmcode und führt diese Zeile aus.

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

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

- a) test.o
- b) test.exe
- c) test.lib
- d) test.zip
- e) **test.i**

1.5 Was ist keine höhere Programmiersprache?

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

1.6 Welche Dateieindung 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 unsigned char Variablen darstellen kann?

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

1.8 Wie sieht die Definition einer modulglobalen Variablen j in einer c-Datei aus?

- a) int j; innerhalb einer Funktion
- b) static int j; innerhalb einer Funktion
- c) int j; außerhalb einer Funktion
- d) static int j; außerhalb einer Funktion**
- e) #define j 3.1415 außerhalb einer Funktion

1.9 Welchen Datentyp gibt es in C89 **nicht**?

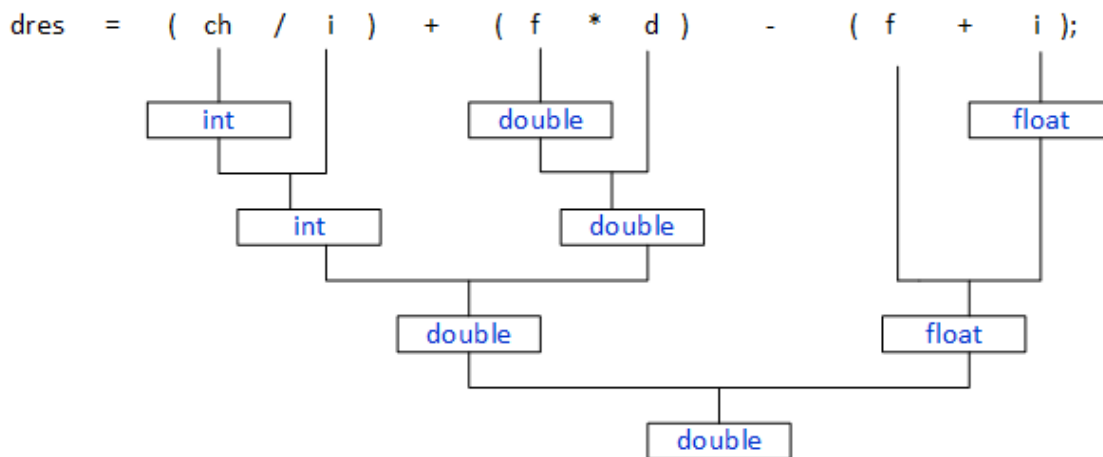
- a) unsigned char
- b) boolean**
- c) unsigned long int
- d) char *
- e) double

1.10 Gegeben sei die Zahl 4811. Es handelt sich **nicht** um eine Zahl aus dem:

- a) Hexadezimalsystem
- b) Dezimalsystem
- c) Zwölfersystem (Duodezimalsystem)
- d) Oktalsystem**

Aufgabe 2:**10 Punkte****2.1** Vervollständigen Sie die folgende Abbildung aus dem Skript – Thema casts! (**4 Punkte**)

```
char ch;
int i;
float f;
double d, dres;
```



½ Punkt pro richtiger Eintrag

2.2 Wandeln Sie die folgenden positiven Zahlen (unsigned) in das andere Zahlensystem um: (**4 x 1 Punkt**)

$$\begin{aligned}
 1111110101_2 &= 1765_8 \\
 1111110101_2 &= 3F5_{16} \\
 11_{19} &= 20_{10} \\
 65_{10} &= 1000001_2
 \end{aligned}$$

2.3 Wie wird der Wert -1 im Zweierkomplement (8 Bit) dargestellt? (**2 Punkte**)

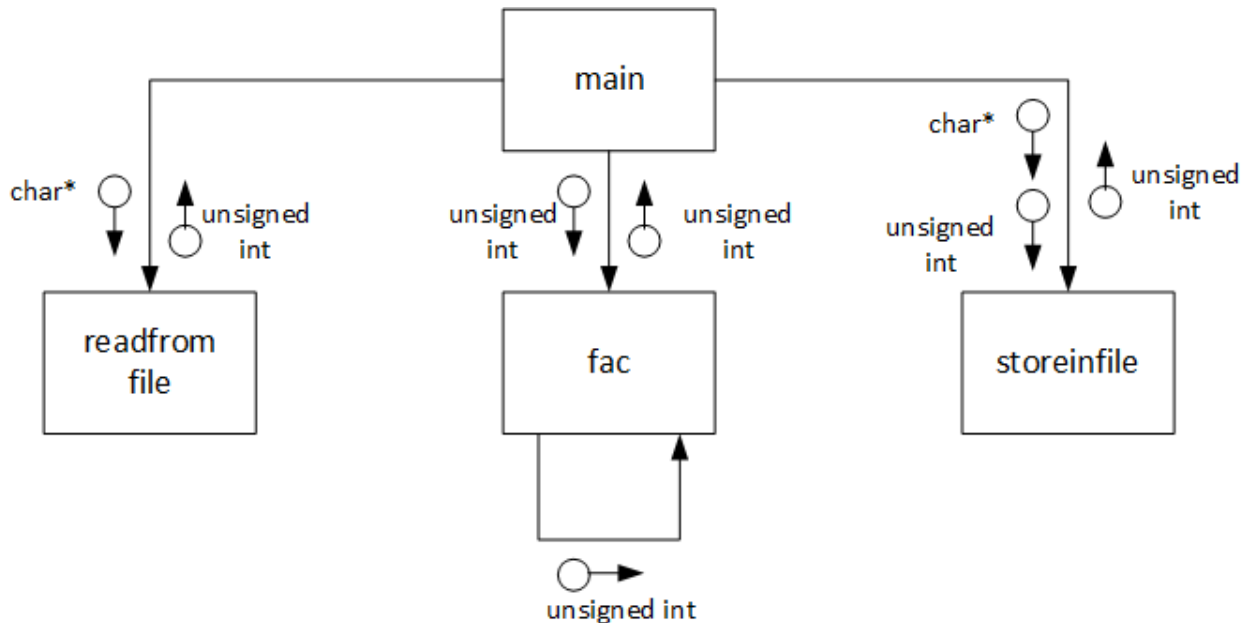
1	1	1	1	1	1	1	1
MSB							LSB

Most Significant Bit

Least Significant Bit

Aufgabe 3:**10 Punkte**

3.1 Das folgende Diagramm enthält einen Fehler! Markieren Sie den Fehler im Diagramm!
(Nur mit Begründung) (8 Punkte)



Begründung:

Die Funktion `fac` hat einmal einen Rückgabewert vom Typ `unsigned int` und einmal ein Rückgabewert vom Typ `void` (rekursiver Aufruf). Dies ist in C nicht möglich, da eine Funktion entweder

- Einen Rückgabewert (ein bestimmter Datentyp)

EXOR

- Keinen Rückgabewert hat (`void`)

Im Diagramm ist dies nicht konsistent

3.2 Deklarieren Sie die Funktionen aus obiger Abbildung (inklusive `main`)! (2 Punkte)

```
int main (int argc, char * argv[])
```

Alternative: `int main (void)`

```
unsigned int readfromfile (char *);
```

```
unsigned int storeinfile (char *, unsigned int); // Reihenfolge vertauschbar
```

```
unsigned int fac(unsigned int);
```

Alternative: `void fac(unsigned int);` // da Fehler im Diagramm

Jeweils einen halben Punkt für eine vollständig richtige Deklaration

Aufgabe 4: (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 tpfs(void)
{
```

```
    int a = 2;
    int b = 4;
    unsigned short int iarray[SIZE][SIZE];
    union quad myquad;
    char szbestcourse[12] = "Ing.-Inf.";
    unsigned char uc1;
    unsigned char uc2;
```

```
    printf("%d\n", sizeof(union quad));           // 4 (1P)
    printf("%d\n", sizeof(struct element));        // 8 (1P)
    printf("%d\n", sizeof(PSTRUCTELEMENT));        // 4 (1P)
    printf("%d\n", sizeof(struct bits));           // 2 (1P)
    printf("%d\n", sizeof(long long int));         // 8 (1P)
    printf("%d\n", sizeof(double*));              // 4 (1P)
    printf("%d\n", SUB(a,b));                      // -2 (1P)
    printf("%d\n", 2*SUB(a,b));                   // 0 (1P)
    printf("%d\n", SUB(a,++b));                   // -3 (1P)
    printf("%d\n", sizeof(iarray));               // 18 (1P)
```

```
    strcpy(myquad.sztext, "FH0");
    myquad.sival[0] = 0x4945;
    myquad.sival[1] = 0x002B;
    printf("%s\n", myquad.sztext);                // _EI+ (2P)
```

```
    printf("%d\n", sizeof(szbestcourse)
        - strlen(szbestcourse));                 // 3 (2P)
    printf("%s\n", strtok(szbestcourse, "."));    // _Ing (2P)
```

```
    uc1 = 0x96;
    uc2 = 0xFF;
    printf("%x\n", (unsigned char) ((~uc1) ^ uc2)); // _96 (2P)
    printf("%x\n", (unsigned char) ((uc2>>8) &
        ((uc1|(uc2<<2)))));                      // _0 (2P)
```

```
}
```

```
union quad
{
    char sztext[4];
    short int sival[2];
};
struct element
{
    char * pdata;
    struct element * pnext;
};
struct bits
{
    unsigned char ucv1 : 4;
    unsigned char ucv2 : 5;
};
#define SIZE 3
#define SUB(a,b) a-b
typedef struct element * PSTRUCTELEMENT;
```

MyVars.h

Aufgabe 5: (15 Punkte)

Gegeben sei die C-Funktion hoo, wobei n die Anzahl der Elemente in array ist.

```
void hoo(int * array, unsigned int n) // n Elements in array
{
    unsigned int c, d, position;
    int swap;

    for (c = 0; c < (n-1); c++)
    {
        position = c;
        for (d = c + 1; d < (n-1); d++) //-1 muss weg
        {
            if (array[position] > array[d])
                position = d;
        }
        if (position != c)
        {
            swap = array[c];
            array[c] = array[position];
            array[position] = swap;
        }
    }
}
```

5.1. Welche Aufgabe hat die C-Funktion hoo? (2 Punkte)

Sortiert ein Array mit Insertion Sort

5.2 Die Funktion hoo enthält einen Fehler. Korrigieren Sie diesen Fehler im obigen Programmcode! (13 Punkte)

Aufgabe 6 (15 Punkte)

Schreiben Sie eine **rekursive** Funktion foo, die Eingaben von der Tastatur einliest bis das Zeichen ‚e‘ eingegeben wird. Danach werden alle eingegebenen Zeichen in umgekehrter Reihenfolge inklusive dem ‚e‘ auf der Konsole ausgegeben:

Beispiel: Eingabe *Halloe* Ausgabe nachdem ‚e‘ gedrückt wurde *eollaH*

```
void foo (void)
{
    int ich;

    ich = _getch();

    if (ich != 'e')
        foo();

    printf("%c", ich);
}
```

Aufgabe 7 (20 Punkte)**7.1.** Implementieren Sie die Funktion checkvalid. (8 Punkte)

Übergabewert: char* p 32 stellige Binärzahl als Zeichenkette, mit 0x00 abgeschlossen!

Rückgabewert: int -1 wenn p NULL ist

-2 wenn p keine 32 Zeichen enthält

-3 wenn p 32 Zeichen enthält, aber andere Zeichen als '0' und '1'

0 wenn p ungleich NULL ist und 32 '0' oder '1' Zeichen enthält.

```
int checkvalid (char *p)
{
    int iret;
    int i;

    if (p == NULL)
    {
        iret = -1;          // 1 Punkt
    }
    else
    {
        if (strlen(p) == 32) // 1 Punkt
        {
            iret = 0;

            for (i = 0; i < 32 /* && iret == 0 */; i++) // 2 Punkte
            {
                if (p[i] != '1' && p[i] != '0') // 3 Punkte
                    iret = -3;
            }
        }
        else
        {
            iret = -2; // 1 Punkt
        }
    }
    return iret;
}
```


7.2 Implementieren Sie die Funktion `convert32stringtounsignedint` (12 Punkte)

Übergabewert: char* p

Rückgabewert: unsigned int

0, falls `checkvalid() != 0` ist

Ansonsten konvertiert die Funktion eine Zeichenkette (32 Zeichen), welche eine 32-stellige Binärzahl als Zeichenkette darstellt in den entsprechenden

unsigend int Wert

Beispiel: "00000000000000000000000000001010" -> unsigned int Wert 10

```

unsigned int convert32stringtounsignedint (char *p)
{
    unsigned int Wert = 0;

    if (checkvalid(p) != 0)
    {
        Wert = 0; // 1 Punkt
    }
    else
    {
        Wert = 0; // 1 Punkt

        for (j = 0; j < 32; j++) // Schleife 2 Punkt
        {
            // Algorithmus 8 Punkte
            // Lösung 1
            //Wert = Wert + (p[31 - j] - 48)*pow(2,j);

            // Lösung 2
            //Wert = Wert | ((p[31 - j] - 48) << j);

            //Natürlich auch länger und verständlicher
            //Schleife könnte auch von 31 bis 0 laufen.
            if (p[31 - j] == '1')
            {
                uib = 1;
            }
            else
            {
                uib = 0;
            }
            Wert = Wert + uib * pow(2, j);
        }
    }
    return Wert;
}

```

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