

KSP - Tutorium

Test 3 Africa's Geek



KSP

Test für die Klausurvorbereitung – KSP

Nachname:	
Vorname:	
Betreuer:	Donchi Fofack Donald

	Max. Punktzahl	erreicht
Aufgabe 1	20	
Aufgabe 2	5	
Aufgabe 3	15	
Aufgabe 4	20	
Aufgabe 5	20	
Aufgabe 6	20	
Wissensfragen(Bonus punkte)	10	
Gesamt	110	

Aufgabe 1: (20 Punkte)

Gegeben ist folgender Code:

```
int *(*h)(char *s);
double f(int i, char c);
int main(int argc, char const *argv[])
{
    signed short i = 4;
    unsigned char a[30];
    float *p;
    long *b[20];
    int z[5] = {3, -1, 0, 4, 2};
    int *y = z + 2;
    int s = *(z + 2);
    int k = *z + 2;
    printf("sizeof(i) = %ld\n", sizeof(i));
    printf("sizeof(a) = %ld\n", sizeof(a));
    printf("sizeof(p) = %ld\n", sizeof(p));
    printf("sizeof(b) = %ld\n", sizeof(b));
    printf("sizeof(z) = %ld\n", sizeof(z));
    printf("sizeof(y) = %ld\n", sizeof(y));
    printf("sizeof(s) = %ld\n", sizeof(s));
    printf("sizeof(k) = %ld\n", sizeof(k));
    printf("z[z[z[4]]] = %d\n", z[z[z[z[4]]]);
    return 0;
}
```

a) Ergänzen Sie die umgangssprachlichen Erklärungen zu obenstehenden Deklarationen. Zu Vorbelegungen brauchen Sie nichts schreiben! (benutzen Sie dabei deutsche Begriffe): (11 Punkte)

h ist ...

f ist ...

main ist ...

i ist ...

a ist ...

p ist ...

b ist ...

z ist ...

y ist ...

s ist ...

k ist ...

b) welche Ausgabe erscheint bei der Ausführung des Programms?

NB: Der Architektur des Zielsystems X86_64 (9 punkte)

.....

Aufgabe 2: (5 Punkte)

Wir haben vier c-datei: **main.c**, **helper.c**, **fack.c** und **mod.c**:

Welcher Befehl benötigt man, damit ein ausführbares Programm (**main**) aus den beiden C-Quellcode erstellt wird?

Dabei soll alle Warnungen ausgegeben werden, das Debug-symbol soll im Programm vorhanden sein, das Programm soll nach c99 standart kompiliert werden, das Programm soll pedantisch an Standart halten.

.....

Aufgabe 3: (15 Punkte)

Ergänzen Sie bitte die Kommentar bei A, B, C, D und E wie bei dem Beispiel (6 Punktes)

Beispiel:

```
typedef struct person
{
    int *b;//8
    int a;//4
}Person;//min_größe=12 , max_größe= 16
```

```
//A
typedef struct person
{
    int *b;//?
    int a;//?
    int c;//?
}PersonA;//min_größe=? , max_größe=?
```

```
//B
typedef struct person
{
    int *b;//?
    double a;//?
    char c;//?
}PersonB;//min_größe=? , max_größe=?
```

```
//C
typedef struct person
{
    short b;//?
    int a;//?
    char c;//?
}PersonC;//min_größe=? , max_größe=?
```

```
//D
typedef union person
{
    short b;//?
    char c;//?
}PersonD;//max_größe=?
```

```
//E
typedef struct person
{
    int a; //?
    union d //?
    {
        int e; //?
        double f; //?
    };

    char c; //?
} PersonE; //min_größe=? , max_größe=?
```

Aufgabe 4: (20 Punkte)

a) Schreiben Sie einen Makro TEST_EQ(x,y,n), der genau dann “wahr” (also 1) abliefern, wenn in x und y das n-te Bit übereinstimmt. Dabei sollen Bits vom LSB beginnend mit 0 nummeriert sein. (10 Punktes)

b) Es sei “unsigned int n = 0xA890” gegeben. Dann wird die Anweisung $n \&= n - 1$; ausgeführt. Geben Sie den Wert von n nach der Anweisung an (hexadezimal). (5 Punktes)

c) Welchen Wert gibt die untenstehende Funktion zurück, wenn sie mit $x=0xA890$ aufgerufen wird? (5 Punktes)

```
int f(unsigned int x){
    int n = 0;
    while(x){
        if( x & 1) n++;
        x >>= 1;
    }

    return n;
}
```

Aufgabe 5: (20 Punktes)

1- Schreiben Sie in Ninja Assembler ein Programm, welches die Fakultät von 10 mit einer Schleife berechnet. (10 Punktes)

2- Schreiben Sie eine Funktion in Ninja-Assembler, die ein Programm, welches die Fakultät einer beliebigen Zahl berechnet. (10 Punktes)

NB: fakultät(3) = 3 * 2 * 1 = 6;

Aufgabe 6: Pointer Trainer(20 Punktes)

Beispiel: Der Wert von z sei gegeben als 0x4EDED08B. Markieren Sie die Bytes, die durch den Ausdruck *y dereferenziert werden.

```
Typen
char: 1 Byte
char *: 4 Byte

char *z = "at laoreet tur";
char *y = &z[12];
```

0x4EDED040	FA 6D 61 67	6E 61 2C 20	70 75 6C 76	69 6E 61 72	.magna, pulvinar
0x4EDED050	20 00 09 7C	11 D9 41 35	3D A9 63 A1	50 7C 9B 84A5=.c.P ..
0x4EDED060	F3 C3 85 23	C6 C4 1A CC	41 EC 4F ED	00 7B 88 DD	...#....A.0..{..
0x4EDED070	DC 68 BF D7	C6 74 2E 20	53 75 73 70	00 CA E2 0E	.h...t. Susp....
0x4EDED080	90 1C C9 85	66 94 E2 3D	9A 26 FD 61	74 20 6C 61f..=.&.at la
0x4EDED090	6F 72 65 65	74 20 74 75	72 00 D5 57	31 E4 D1 2E	oreet tur..Wl...

Aufgabe 1: Der Wert von z sei gegeben als 0x105E2DFB. Markieren Sie die Bytes, die durch den Ausdruck *y dereferenziert werden. (5 Punkte)

Typen
char: 1 Byte
char *: 4 Byte

```
char *z = "ipsum dolor";
char *y = &z[2];
```

0x105E2DE0	6E 65 71 75	65 2C 20 00	71 63 6F 6E	73 65 63 74	neque, .qconsect
0x105E2DF0	65 74 75 72	20 61 64 69	70 00 2B 69	70 73 75 6D	etur adip.+ipsum
0x105E2E00	20 64 6F 6C	6F 72 00 8F	9D 92 22 61	09 C9 B6 78	dolor...."a...x
0x105E2E10	6D 11 DB 47	A8 9B EB 43	01 8F 56 B6	EC AA 4B E3	m..G...C..V...K.
0x105E2E20	CB F0 43 B7	E9 22 39 B1	06 6B 2D 0A	DA 87 A9 E0	..C.."9..k-.....
0x105E2E30	3F 01 AB E1	25 C6 CF AD	0F FA 5D 29	2D BB 59 87	?...%.....])-.Y.

Aufgabe 2: Markieren Sie die Bytes, die durch den Ausdruck *y dereferenziert werden. (5 Punkte)

Typen
int: 4 Byte
int *: 4 Byte

```
int *z = (int *) 0x61946F58;
int *y = &z[9];
```

0x61946F50	1C D6 4C BF	10 CC C9 D8	F8 1C 32 D5	26 EB EC 01	..L.....2.&...
0x61946F60	89 1D B5 03	BD CE 36 08	3C 43 23 B9	B4 90 06 A46.<C#.....
0x61946F70	A1 2E BF E8	B1 89 35 1E	59 AE 3B C5	BF 92 45 4C5.Y.;...EL
0x61946F80	6C 23 33 BB	B6 4D 7C A7	B5 ED F4 68	B5 26 89 05	l#3..Mh.&..
0x61946F90	A1 17 70 55	72 F5 44 61	90 3E C8 3A	3A EF 32 F3	..pUr.Da.>...2.
0x61946FA0	A6 85 7C 77	26 D7 8A 23	08 FC 97 92	E4 87 9A 39	.. w&..#.....9

Aufgabe 3: Markieren Sie die Bytes, die durch den Ausdruck `*y` dereferenziert werden. (5 Punktes)

```
Typen
int: 4 Byte
int *: 4 Byte
char: 1 Byte
char *: 4 Byte

int *z = (int *) 0x554ABA24;
char *y = (char *) (z+13);
```

0x554ABA00	DC F7 AE B1	51 69 E8 BE	CA AE 4D 03	C2 A9 D4 3CQi....M....<
0x554ABA10	CC 32 C6 50	EC A7 E3 67	6A 26 FA 6D	F2 36 3C B5	.2.P...gj&.m.6<.
0x554ABA20	5B F6 68 66	01 AC 8E A3	E9 10 ED 33	24 A7 19 EE	[.hf.....3\$...
0x554ABA30	FA B4 7E 4C	37 58 A3 75	C1 31 C6 54	E2 00 77 2A	..~L7X.u.1.T..w*
0x554ABA40	E0 18 A9 B1	20 25 69 5F	49 C9 4C 58	DE 97 6C 85 %i_I.LX..l.
0x554ABA50	E3 2C FF 30	3B F8 A2 90	1F 30 17 00	FB 3C 31 31	.,.0;....0...<11

Aufgabe 4: Der Wert von `z` sei gegeben als `0x498EA51D`. Markieren Sie die Bytes, die durch den Ausdruck `*y` dereferenziert werden. (5 Punktes)

```
Typen
char: 1 Byte
char *: 4 Byte

char *z = "is. Integer vel
mol";
char *y = z+11;
```

0x498EA500	E9 97 48 69	AD 12 73 76	51 A3 CC 31	2F 2F 90 F3	..Hi..svQ..1//..
0x498EA510	D4 EF 32 6A	2B 2F B3 67	7B 86 6C 60	61 69 73 2E	..2j+/.g{.l`ais.
0x498EA520	20 49 6E 74	65 67 65 72	20 76 65 6C	20 6D 6F 6C	Integer vel mol
0x498EA530	00 FC 1D DB	43 A6 06 0C	37 D0 AF 97	E2 FE BB 42C...7.....B
0x498EA540	0E B6 20 49	6E 74 65 67	65 72 20 76	65 6C 20 6D	.. Integer vel m
0x498EA550	6F 6C 00 20	73 65 64 20	6D 65 00 72	71 2F 5E 04	ol. sed me.rq/^.