KSP - Tutorium

Test 1
Africa‘s Geek

KSP

Donchi Fofack Donald

Test für die Klausurvorbereitung – KSP

Nachname:	
Vorname:	
Betreuer:	Donchi Fofack Donald

	Max. Punktzahl	erreicht
Aufgabe 1	20	
Aufgabe 2	8	
Aufgabe 3	5	
Aufgabe 4	8	
Aufgabe 5	8	
Wissensfragen	20	
Gesamt	69	

Aufgabe 1:

Gegeben ist folgender Code:

```
5   int main(int argc, char const *argv[])
6   {
7       int a  = 5;
8       long long b;
9       char c = 'H';
10      char str[] = "Hallo Welt";
11      int d = 185;
12      double e = 12.6;
13      int x[] = {-3, 17, 0, 1, -7, -2, 8, 50, -29, -3, 7};
14      int *p = &x[6];
15      int *q = x;
16      int **r = &p;
17      printf("Sizeof(a) = %ld Bytes\n", sizeof(a));
18      printf("Sizeof(b) = %ld Bytes\n", sizeof(b));
19      printf("Sizeof(c) = %ld Bytes\n", sizeof(c));
20      printf("Sizeof(str) = %ld Bytes\n", sizeof(str));
21      printf("Sizeof(d) = %ld Bytes\n", sizeof(d));
22      printf("Sizeof(e) = %ld Bytes\n", sizeof(e));
23      printf("Sizeof(x) = %ld Bytes\n", sizeof(x));
24      printf("Sizeof(p) = %ld Bytes\n", sizeof(p));
25      printf("Sizeof(q) = %ld Bytes\n", sizeof(q));
26      printf("Sizeof(r) = %ld Bytes\n", sizeof(r));
27      return 0;
28  }
```

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

a ist …

b ist …

c ist …

d ist …

e ist …

x ist …

p ist ...

q ist …

r ist …

main ist ...

b) welche Ausgabe erscheint bei der Ausführung des Programms?
NB: Der Architektur des Zielsystems X86_64
Sizeof(a) = … Bytes
Sizeof(b) = … Bytes
Sizeof(c) = … Bytes
Sizeof(str) = … Bytes
Sizeof(d) = … Bytes
Sizeof(e) = … Bytes
Sizeof(x) = … Bytes
Sizeof(p) = … Bytes
Sizeof(q) = … Bytes
Sizeof(r) = … Bytes

Aufgabe 2:

Damit ein ausführbares Programm aus einem C-Quellcode erstellt wird, benötigt man folgender Befehl: gcc -g -pedantic -std=c99 -Wall -o main main.c.
Welche Rolle spielt oder was ist :
***gcc:**
***-g:**
***-pedantic:**
***-std=c99:**
***-Wall:**
***-o:**
***main:**
***main.c:**

Aufgabe 3:

Definieren Sie folgende Makros:

a) FUNCTION1:(a+b+c) / (a*b*c)

b) FUNCTION2: x^3+3x^2+3x+1

c) FUNCTION3: $2x – 4$

d) FUNCTION4: $x – 1$

e) FUNCTION5: 100

Aufgabe 4:

Gegeben Seien die folgenden Header add.h, mul.h und die Datei main.c

add.h	mul.h
…… …... void add(int a, int b); …….	…… …... void mul(int a, int b); …….
main.c	

……
…….
 	
int main(int argc, char *argv[]){
  add(6,9);
  mul(10,7);
}
 	
	Frage: Verhinden Sie bitte die Mehrfachinclusion!!

Aufgabe 5:

→ Schreiben Sie bitte folgende Codestücke in Ninja-Assembler:

a) auswertung von 2*3+5

b) writeInteger((3+4) * (10 – 6));
   writeCharacter(‘\n);

c) auswertung von (-12 * (3-2) / 2) % 5

d) auswertung von ((15 / 3) + (2%7))

e) auswertung von 45*2

→ Zeichen Sie bitte noch den Stackinhalt zu a), b) und e)