ohm Technische Hochschule Nürnberg

Informatik 2/A

Übungen

Wintersemester 2023/2024
Prof. Dr. Anja Freudenreich
Prof. Dr. Jan Paulus
Prof. Dr. Enrico Schröder

ohm

Übungen

Geben Sie Definitionen an für:

1. einen Zeiger ptr auf eine char Variable:
2. ein Array arr mit 100 Elementen vom Typ double:
3. einen Zeiger zarr, der auf ein int-Array der Größe 130 zeigt:
4. ein Array ptrarr mit 12 double-Zeigern:
5. Ein String-Array mit 5 Elementen, wobei jedes Element eine Zeichenkette für einen Wochentag darstellt.
6. Eine Struktur "Auto" mit den Mitgliedern "Marke" (char-Array), "Baujahr" (int) und "Preis" (double).
7. Einen Zeiger, der auf eine dynamisch allokierte Instanz dieser Struktur zeigt.
8. Definieren Sie eine einfach verkettete Liste von „Auto“-Strukturen. Die Listenelemente sollten einen Zeiger auf eine Auto-Struktur sowie einen Zeiger auf das nächste Listenelement enthalten
9. Ein Array von 8 Zeigern auf Funktionen, wobei jede Funktion einen unterschiedlichen Rückgabetyp und Parameter hat.

ohm

Übungen

Geben Sie Definitionen an für:

1. einen Zeiger ptr auf eine char Variable:

```
char *ptr;
```

2. ein Array arr mit 100 Elementen vom Typ double:

```
double arr[100];
```

3. einen Zeiger zarr, der auf ein int-Array der Größe 130 zeigt:

```
int myarr[130];
int *zarr = &myarr;
```

4. ein Array ptrarr mit 12 double-Zeigern:

```
double *ptrarr[12];
```

ohm

Übungen

Geben Sie Definitionen an für:

5. Ein String-Array mit 5 Elementen, wobei jedes Element eine Zeichenkette für einen Wochentag darstellt.

```
char *wochentage[5] = {"Montag", "Dienstag", "Mittwoch",
                       "Donnerstag", "Freitag"};
```

6. Eine Struktur "Auto" mit den Mitgliedern "Marke" (char-Array), "Baujahr" (int) und "Preis" (double).

```
struct Auto {
 char Marke[50];
   int Baujahr;
 double Preis;};
```

Übungen

Geben Sie Definitionen an für:

7. Einen Zeiger, der auf eine dynamisch allokierte Instanz dieser Struktur zeigt.

```
struct Auto *autoPtr = malloc(sizeof(struct Auto));
```

8. Definieren Sie eine einfach verkettete Liste von „Auto“-Strukturen. Die Listenelemente sollten einen Zeiger auf eine Auto-Struktur sowie einen Zeiger auf das nächste Listenelement enthalten

```
struct ListNode {
struct Auto *autoData;
struct ListNode *next;};
```

Übungen

Geben Sie Definitionen an für:

9. Ein Array von 8 Zeigern auf Funktionen, wobei jede Funktion einen unterschiedlichen Rückgabetyp und Parameter hat.

```c
// Beispiel für Funktionen
int func1(int a) { return a * 2; }
double func2(double b) { return b / 2.0; }
void func3(char c) { printf("Character: %c\n", c); }

// Array von Zeigern auf Funktionen
typedef void (*FunctionPointer)();
FunctionPointer funcArray[8] = {func1, func2, func3, /* ... weitere Funktionen hier ... */};
```

Übungen

Was gibt das folgende Programm aus?

```c
#include <stdio.h>
int main()
{
   int eins = 1, zwei = 2, *zeiger1, *z2;
   char c[6] = "Hello";

   zeiger1 = &zwei;
   *zeiger1 = 5;
   z2 = zeiger1;
   (*c)++;

   printf("*zeiger1 = %d\n", *zeiger1); /* *zeiger1 = _____ */
   printf("*z2 = %d\n", *z2); /* *z2 = _____ */
   printf("*c = %s\n", c); /* *z2 = _____ */

   z2 = &eins;
   eins = 18;

   printf("*zeiger1 = %d\n", *zeiger1); /* *zeiger1 = _____ */
   printf("*z2 = %d\n", *z2);           /* *z2 = _____ */

   *zeiger1 = 30;
   *z2 = 12;
   *zeiger1 /= *z2;
   z2 = zeiger1;
   *zeiger1 += *z2;

   printf("eins = %d\n", eins);    /*  eins = _____ */
   printf("zwei = %d\n", zwei);    /*  zwei = _____ */

   return 0;
}
```

Übungen

Was gibt das folgende Programm aus?

```
*z1 = 5
*z2 = 5
*c = Iello
*z1 = 5
*z2 = 18
eins = 12
zwei = 4
```

```c
#include <stdio.h>
int main()
{
   int eins = 1, zwei = 2, *zeiger1, *z2;
   char c[6] = "Hello";

   zeiger1 = &zwei;
   *zeiger1 = 5;
   z2 = zeiger1;
   (*c)++;

   printf("*zeiger1 = %d\n", *zeiger1); /* *zeiger1 = _____ */
   printf("*z2 = %d\n", *z2); /* *z2 = _____ */
   printf("*c = %s\n", c); /* *z2 = _____ */

   z2 = &eins;
   eins = 18;

   printf("*zeiger1 = %d\n", *zeiger1); /* *zeiger1 = _____ */
   printf("*z2 = %d\n", *z2);           /* *z2 = _____ */

   *zeiger1 = 30;
   *z2 = 12;
   *zeiger1 /= *z2;
   z2 = zeiger1;
   *zeiger1 += *z2;

   printf("eins = %d\n", eins);    /*  eins = _____ */
   printf("zwei = %d\n", zwei);    /*  zwei = _____ */

   return 0;
}
```

Autoren / Impressum

- **Autoren**

  Prof. Dr.-Ing. Jan Paulus, Prof. Dr. Enrico Schröder, Prof. Dr. Anja Freudenreich

- **Impressum**

  Prof. Dr. Anja Freudenreich
  Fakultät Elektrotechnik Feinwerktechnik Informationstechnik,
  Wassertorstraße 10
  904489 Nürnberg, Germany
  E-mail : anja.freudenreich@th nuernberg.de