Informatik 2 – Probeklausur

Kombinierte Klausur: Teil A und Teil B

Wintersemester 2023/2024

Aufgabenübersicht

Teil A – C-Programmierung: (60 BE)

Aufgabe 1: Definitionen (8 BE)

Wie werden folgende Objekte definiert:

a) Ein **Zeiger** pdbl auf eine double Variable:

_____

b) Ein **Array** arr mit 100 Elementen vom Typ int:

_____

c) Ein **Zeiger** parr, der auf ein char-Array der Größe 150 zeigt:

_____

d) Ein **Array** words mit 20 Elementen vom Typ char:

_____

e) Ein **Zeiger** arrptr, der auf ein Zeigerarray der Größe 20 mit Integerwerten zeigt:

_____

f) Ein **Array** zgrarray mit 20 int-Zeigern:

_____

g) Ein **Zeiger** ptr2Darr, der auf ein 2D-int-Array mit 20 Zeilen und 50 Spalten zeigt:

_____

h) Ein **Array** a3Darray, das die Inhalte der Fächer eines Wandregals darstellen soll; das Regal ist 5 Fächer hoch, 10 Fächer breit und jedes Fach ist 20 Messeinheiten tief:

_____

Aufgabe 2: Programmausgabe (5 BE)

Was gibt folgendes Programm aus:

```
#include <stdio.h>
static int x = 20;

void foo(int z) {
	static int x = 0, y = 1;
	x += z;
	y *= x;
	printf("x = %d\n", x);
	printf("y = %d\n", y);
}

int main(void){
	int y = 0;
	x++;
	printf("x = %d\n", x);	/* x = _____ */
	printf("y = %d\n", y);	/* y = _____ */
	{
		int x = 100;
		printf("x = %d\n", x); /* x = _____ */
		printf("y = %d\n", y); /* y = _____ */
	}
	foo(17);			/* x = _____  y = _____ */
	foo(35);			/* x = _____  y = _____ */
	printf("x = %d\n", x);	/* x = _____ */
	printf("y = %d\n", y);	/* y = _____ */
	return 0;
}
```

Aufgabe 3: Speicherbelegung (5 BE)

Was gibt das folgende Programm aus, wenn für Zeiger 8 Bytes und für int-Variablen 4 Bytes verwendet werden:

```
#include <stdio.h>

void fkt1(int *ptr) {
     printf("1: %d\n", sizeof(ptr)+2);
}

void fkt2(int arr[]) {
     printf("2: %d\n", sizeof(arr)+5);
}

void fkt3(int arr[200]) {
     printf("3: %d\n", sizeof(arr)+3);
}

void fkt4(int (*mat)[50]) {
     printf("4: %d\n", sizeof(*mat)+6);
}

void fkt5(int mat[100][50]) {
     printf("5: %d\n", sizeof(mat)+2);
}

int main(void) {
   int *ptr,
       a[200],
       z[100][50];

   fkt1(ptr);   /* ____________________ */
   fkt2(a);     /* ____________________ */
   fkt3(a);     /* ____________________ */
   fkt4(z);     /* ____________________ */
   fkt5(z);     /* ____________________ */

   return 0;
}
```

Aufgabe 4: Fehlersuche (12 BE)

Besitzen die folgenden Codestücke Fehler oder nicht und wenn ja um welche Art von Fehler handelt es sich?

a)

```
int z;
int foo (int x){
     z = 0;
     return 4;
```

☐ fehlerfrei ☐ Compilerfehler ☐ Laufzeitfehler ☐ Fehler

b)

```
char *d, dest[10] = { 0 };
char *s = "Text";
while (*s)
  *d++ = *s++;
*d = 0;
```

☐ fehlerfrei ☐ Compilerfehler ☐ Laufzeitfehler ☐ Fehler

c)

```
char array[100];
strcpy (array, ”Hans”);
array += 4;
strcpy (array, "wurst");
```

☐ fehlerfrei ☐ Compilerfehler ☐ Laufzeitfehler ☐ Fehler

d)

```
char *src = "Text";
char dest[10];
char *d = dest, *s = src;
while (*d++ = *s++)
  ;
```

☐ fehlerfrei ☐ Compilerfehler ☐ Laufzeitfehler ☐ Fehler

e)

```
#define MAX   100
int array[MAX], i;
for (i = 1; i <= MAX; i++)
    array[i] = i*i;
```

☐ fehlerfrei ☐ Compilerfehler ☐ Laufzeitfehler ☐ Fehler

f)

```
char array[100];
strcpy(array, ”Hans”);
free(array+5);
```

☐ fehlerfrei ☐ Compilerfehler ☐ Laufzeitfehler ☐ Fehler

Aufgabe 5: Makros (8 BE)

a) Der Code ist zu ergänzen, sodass die folgende Ausgabe geliefert wird:

```
HOCH2_PLUS1(2) * 2 = 10
HOCH2_PLUS1(3) * 2 = 20
HOCH2_PLUS1(4) * 2 = 34
HOCH2_PLUS1(5) * 2 = 52
```

```
#include <stdio.h>
#define HOCH2_PLUS1(___________)   _____________________________

int main(void){
   int a = 10, b = 5, i;
   for (i = 1; i <= 4; i++)
     printf("HOCH2_PLUS1(%d) * 2 = %d\n", i+1, HOCH2_PLUS1(i+1) * 2);
   return 0;
}
```

b) Der Code ist zu ergänzen, sodass die beiden Parameter subtrahiert werden; was ist die Ausgabe des Programms?

```
#include <stdio.h>

#define SUB(_____, _____)   ____________________________________

int main(void){
      int   m = 20,   n = 45;

      m = 2 * SUB( m , n + 1 );

      printf("m = %d\n", m);       /* ____________________ */
      return 0;
}
```

Aufgabe 6: Quadratsumme (8 BE)

Das folgende Programm quadsum.c soll eine Zahl n einlesen und dann die Summe aller Quadratzahlen bis zu dieser Zahl ausgeben. Die Berechnung soll rekursiv stattfinden und die einzelnen Quadratzahlen ausgeben, die für die Berechnung verwendet werden.

```
#include <stdio.h>

int quadsum(int n){









}

int main(void) {
      int  n;
      printf("Gib eine Zahl ein: ");
      scanf("%d", &n);
      printf("%d\n", quadsum(n));
      return 0;
}
```

Mögliche Beispiele für den Ablauf des Programms:

```
Gib eine Zahl ein: 4
16 + 9 + 4 + 1 = 30
```

```
Gib Zahl ein: 8
64 + 49 + 36 + 25 + 16 + 9 + 4 + 1 = 204
```

```
Gib Zahl ein: 12
144 + 121 + 100 + 81 + 64 + 49 + 36 + 25 + 16 + 9 + 4 + 1 = 650
```

Aufgabe 7: Strings (8 BE)

Was gibt das folgende Programm aus, wenn es mit den Argumenten **10293** und **40478** aufgerufen und ausgeführt wird?

```
#include <stdio.h>
#include <string.h>

char *ps[] = { "sznm01__0d",
               "uler00xt0r",
               "lofa11is0i",
               "pHOw00si1w"     };

int main(int argc, char *argv[]){
   char **ppc, *pa;
   pa = *++argv;
   while (--argc) {
      while (*pa) {
         ppc = ps + 3;
         do {
            printf("%c", (*ppc)[*pa - '0']);
         } while (ppc-- - ps);
         printf("  ");
         pa++;
      }
      printf("\n");
      pa = *++argv;
   } return 0;
}
```

Ausgabe des Programms:

_____

_____

Aufgabe 8: Sortieren (6 BE)

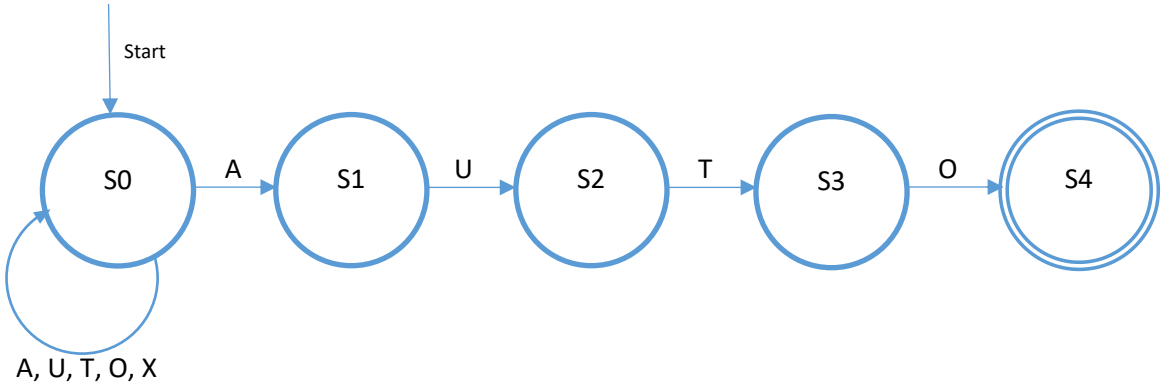
Das folgende Programm soll ergänzt werden, sodass das Array z mithilfe eines Q-Sort Algorithmus aufsteigend sortiert wird:

```
#include  <stdio.h>
#include  <stdlib.h>
#include  <time.h>

#define SIZE 10000
int    i, n;
double z[SIZE];
/* Vergleichsroutine für qsort-Aufruf */




int  main(void) {
   srand(time(NULL));
   for (i=0; i<SIZE; i++)
      z[i] = (double)rand()/1000;
/* qsort-Aufruf */




   return 0;
}
```

Teil B – Automaten: (30 BE)

Aufgabe 1: NEAs und DEAs (9 BE)

Gegeben ist folgender NEA, der die Zeichenkette „AUTO“ erkennt. Der Automat soll in einen DEA konvertiert sowie eine Erreichbarkeitstabelle dazu erstellt werden.



a) Erreichbarkeitstabelle:

Zustand	A	U	T	O	X

b) DEA

Aufgabe 2: Alphabete (7 BE)

a) Ein Automat versteht folgende Sprache:

$$E = \{ w \mid (a^+b)^*c^+ \}$$

Werden folgende Ausdrücke erkannt?

Ausdruck	**erkannt**	**nicht erkannt**
aaaababcc		
abaabaaabbccc		
ccc		
abcabcabc		
aaabaaabacc		

b) Folgende Ausdrück werden erkannt bzw. nicht erkannt:

Erkannte Ausdrücke	Nicht erkannte Ausdrücke
Hallo	Halo
Haalloooo	Haalooo
HalloHaalloo	Haaallo
HaallHaall	HaalloHelloHallo

Welche Regular Expression ist für diese Ausdrücke möglich:

_____

Aufgabe 3: Bäume (14 BE)

Gegeben ist folgende Wahrheitstabelle, aus der ein vollständiger Binärbaum mit Ausgangsbelegung sowie ein ROBDD mit mindestens einem Zwischenschritt entwickelt werden soll:

D	C	B	A	y
0	0	0	0	0
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1
0	0	1	0	0
1	0	1	0	1
0	1	1	0	0
1	1	1	0	1
0	0	0	1	0
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	0
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

Die Ordnung des Baums soll A < B < C < D sein.

a) Vollständiger Binärbaum

b) ROBDD (mindestens einen Zwischenschritt mit aufzeichnen!!)