

Aufgabe 1: (5 Punkte)

Finden Sie alle Syntax Fehler. Folge Fehler sollen nicht berücksichtigt werden.

```

1  int MAIN( int argc, char ** argv )
2  {
3      int a b,F;
4      printf( "Bitte Seite a eingeben: " ),
5      scanf("%d" , "%d", &a , &b );
6      printf( "Bitte Seite b eingeben: " );
7      scanf( "%d", b );
8      F=A*b;
9      print( "Der Flaecheninhalt betraegt F=%d m*m\n", F );
10 }
```

Aufgabe 2: (6 Punkte)

Welche Ergebnisse haben die nachfolgen Ausdrücke, wenn diese jeweils in den printf in Zeile 4 eingesetzt werden.

```

1  int main(int argc, char *argv[])
2  {
3      int i = 1 , j = 2 , n = 0;
4      printf("Ergebnis: %d" , i > 2);
5      return 0;
6  }
```

Ausdruck	Ergebnis
i > 2.....	0
i < 2.....	
i < 2 && n == 0.....	
i.....	
! i.....	
! n.....	
i == 1 && j.....	
i > 2 n == 0.....	
i > 2 j == 2.....	
i && ! n.....	
i == 2 && (j == 1 n == 1)	
(i == 2 && j == 1) n == 1	
i && j && ! n.....	

Aufgabe 3: (7,5 Punkte)

Welche Zeilen werden für welchen Wert von *i* ausgeführt? Die Zuweisung an *i* in Zeile 5 soll gedacht jeweils durch die Werte in der unten stehenden Tabelle gesetzt werden.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i = 8;
6      //scanf("%d" , &i);
7      if( i > 1)
8          printf("i= %d\n" , i);
9      if( i > 1 && i < 6)
10         printf("i= %d\n" , i);
11     if(i > 1 || i == -4)
12         printf("i= %d\n" , i);
13     if(i < 0)
14         if(i > -10)
15             printf("i= %d\n" , i);
16     else printf("i= %d\n" , i);
17     return 0; // Wird immer ausgeführt
18 }
```

Ausgeführte Zeilen					
i	8	10	12	15	16
-20					
-5					
-4					
0					
3					
8					

Aufgabe 4: (5 Punkte)

Geben Sie an, wie die exakte Ausgabe der jeweiligen printf Anweisung aussieht.

Kennzeichnen Sie Leerzeichen mit `_`. Der Zeilenumbruch muss nicht mit angegeben werden.

Anweisung:

Ausgabe:

```
printf("01:%f\n",123.456);
printf("02:%e\n",12.34) ;
printf("03:%6.2f\n",123.4);
printf("04:%9.4f\n",123.4);
printf("05:%6.2f\n\n",-1.2);
printf("06:%06.2f\n",-1.234);
printf("07:% 6.2f\n",1234.) ;
printf("08:%-9.2f\n",1234.) ;
printf("09:%03.2e\n",-12.34) ;
printf("10:%9.1E\n",12.34);
```

Aufgabe 5: (5 Punkte)

Welche Werte haben die #define Konstanten und die Variable i?

```
1  #define A 4711
2  #define B
3  #ifdef A
4      #define N1 17
5      #ifdef B
6          #define N2 2
7      #else
8          #define N2 4
9      #endif
10 #else
11     #define N1 12
12     #define N2 -3
13 #endif
14
15 int i = N1 * N2;
```

A =

B =

N1 =

N2 =

i =

Aufgabe 6: (3 Punkte)

Vervollständigen sie die nachfolgende Tabelle.

Definition	Dimensionen	Größe in Bytes	gültige Indizes
char c[2][4][2]316	$(0..1) \times (0..3) \times (0..1)$
char c[3][3]			
char c[2][2][2][2]			
char c[1][2][1]			
char c['z'-'a'][2]			
char c['z'-'a'][1]			
char c[3][7][2][3]			

Aufgabe 7: (6 Punkte)

Wie sehen die originalen C-Datentypen der nachfolgenden typedef's und Deklarationen aus:

```

1  typedef char *STRING;
2  typedef STRING *STR_PTR;
3
4  typedef char LABEL[5];
5  LABEL lable, *l_ptr;
6
7  typedef int VEC[3] , *XP;
```

Zeile	Name	resultierender C-Typ	Beschreibung
1	STRING	char *	Zeiger auf char
	STR_PTR		
4	LABEL		
5	lable		
	l_ptr		
7	VEC		
	XP		