Aufgabe 1:

a)

```
int main(void)
{
    return 0;
}
```

3/4

b)

```
#include <stdio.h>
int main(void)
{
    printf("Hallo");
    return 0;
}
```

c)

```
#include <stdio.h>

int main (void)
{
    int x;
    x = 3;
    printf("%i",x);
    return 0;
}
```

d)

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

int main(void)
{
    int n;
    n = 5;
    printf("%i \n", n);
    return 0;
}
```

Aufgabe 2:

a)

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

int main(void)
{
    int k, m, n;
    k = -1;
    m = rand();
    n = k * m;
    printf("%i",n);
    return 0;
}
```

b)

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

int main(void)
{
    int n;
    n = INT_MIN;
    n = n - 1;
    printf("%i", n);
    return 0;
}
```

c)

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    char c;
    c = '?';
    printf("%i",c);
    return 0;
}
```

d)

```
#include <stdio.h>

int main(void)
{
	double x;
	x = 4.5 * 10e2;
	printf("%i", x);
	printf("\n%f", x);
	return 0;
}
```

e)

```
#include <stdio.h>

int main(void)
{
	double x;
	int k;
	x = 3.9;
	k = 3.9;
	printf("%.10f %.10f", x, (double)k);
	x = x / 2;
	k = k / 2;
	printf("\n%.10f %.10f", x,(double) k);
	return 0;
}
```

f)

```
#include <stdio.h>


int main(void)
{
    int n = 7;
    printf("%.10f\n", (double)n / 4);
    printf("%.10f\n", n / 4.0);
    return 0;
}
```

g)

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

int main(void)
{
    printf("%f\n", cos(5.0));
    printf("%f\n", log(250.0));
    return 0;
}
```

Aufgabe 3

a)

```
#include <stdio.h>

int main(void)
{
	int i;

	printf("Zahl \t|\t Quadrat\n"); /* Formatierung der Tabelle*/
	for(i = 0; i < 24; ++i) {
		printf("%c", '-');
	}
	printf("\n");

	for(i = 0; i <= 100; ++i) {		/* Ausgabe der Tabelle*/
		printf("%i \t|\t %i\n", i, i*i);
	}

	return 0;
}
```

b)

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

int main(void)
{
	int n, i;
	srand(time(NULL));
	n = rand() % 128;


	printf("Zeichen \t|\t ASCII\n");		/* Formatierung der Tabelle*/

	for(i = 0; i < 30; ++i) {
		printf("%c", '_');
	}
	printf("\n");

	for(i = 0; i <= n; ++i) {			/*Ausgabe der Tabelle*/
		if(isalpha(i)) {
			printf("%c \t|\t Ja\n", i);
		} else {
			printf("%c \t|\t Nein\n", i);
		}
	}

	return 0;
}
```

c)

Es werden zufällig zwischen zwei und fünf Zeilen ausgegeben. In jeder neuen Zeile wird ein Paar Nullen (‚00‘) hinzugefügt.

Mögliche Ausgabe des Programms:

00

0000

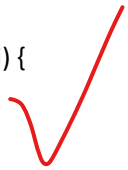
000000

1/1

Aufgabe 4

a)

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

int sum(int w[], int size);
int main(void)
{
	int i;
	int size = 30;
	int w[50];
	srand(time(NULL));

	for(i = 0; i < 50; ++i)
	{
		w[i] = rand();
	}

	printf("%i\n", sum(w, size));
}

int sum(int w[], int size)
{
	int i;
	int k = 0;

	for(i = 0; i < size; ++i) {
		k += w[i];
	}

	return k;
}
```



b)

```
#include <stdio.h>

int array_to_upper(char v[], char w[], int size);
int main(void)
{
	int i;
	char v[12] = {'I', 'n', 'n', 'f', 'o', 'r', 'm', 'a', 't', 'i', 'k', '1'};
	char w[12];
	int size = sizeof(v) / sizeof(v[0]);

	if(array_to_upper(v, w, size) == 0) {
		for(i = 0; i < size; ++i) {
			printf("%c\t", w[i]);
		}


	}

	return 0;
}

int array_to_upper(char v[], char w[], int size)
{
	int i, k;
	for(i = 0; i < size; ++i)
	{
		if((v[i] > 64 && v[i] < 90) || (v[i] > 96 && v[i] < 122))
		{
			if(v[i] > 96 && v[i] < 122)
			{
				w[i] = v[i] - 32;
			}
```

Es soll abgebrochen werden wenn nicht-Buchstaben im array sind. Ihr ignoriert das

```
                else
                {
                        w[i] = v[i];
                }
                k = 0;
        }
        else
        {
                k = 1;
        }
    }

    return k;
}
```

c)

```
#include <stdio.h>

void manipulate_and_print(int v[], int size);

int main(void)
{
    int v[9] = {3, -3, 6, 13, 7, 8, 30, -10, 1028};
    int size = sizeof(v) / sizeof(v[0]);

    manipulate_and_print(v, size);

    return 0;
}

void manipulate_and_print(int v[], int size)
{
    int i;
```

```
	for(i = 0; i < size; ++i) {
		if(v[i] >= 0) {
			switch(v[i] % 3) {
				case 0:
					printf("%i\n", v[i] / 2);
					break;

				case 1:
					printf("%i\n", ++v[i]);
					break;

				case 2:
					printf("%i\n", -v[i] - 2);
					break;

				default:
					printf("fehler");
			}
		} else
			printf("\n");
	}
}
```

d)

```
#include <stdio.h>

void copy_and_transform_digits(char v[], char w[], int size);
int main(void)
{
	int i;
	char v[12] = {'0', '1', '2', '3', '4', '5', 'A', 'B', 'C', 'd', 'E', '9'};
	char w[12];
	int size = sizeof(v) / sizeof(v[0]);
```

```
	copy_and_transform_digits(v, w, size);
	printf("Feld v: ");
	for(i = 0; i < size; ++i) {
		printf("%c\t", v[i]);
	}
	printf("\n");
	printf("Feld w: ");
	for(i = 0; i < size; ++i) {
		printf("%c\t", w[i]);
	}

	return 0;
}

void copy_and_transform_digits(char v[], char w[], int size)
{
	int i;

	for(i = 0; i < size; ++i) {
		if(v[i] > 47 && v[i] < 58) {
			w[i] = 138 - v[i];
		} else {
			w[i] = v[i];
		}
	}

}
```