```c
/*Aufgabe 34a*/

#include <stdio.h>


void increment_if_zero(int *x, int *y);
int main(void)
{
	int a;
	int b;
	int c;

	a = -1;
	b = 5;
	c = 0;

	printf("b vor increment: %i\n", b);
	increment_if_zero(&a, &b);
	printf("increment_if_zero(-1, 5): %i\n", a);
	printf("b nach increment: %i\n", b);

	printf("c vor increment: %i\n", c);
	increment_if_zero(&a, &c);
	printf("increment_if_zero(-1, 0): %i\n", c);
	printf("c nach increment: %i\n", c);



	return 0;
}

void increment_if_zero(int *x, int *y)
{
	if(*y == 0){
		++(*x);
	}
}
```

```c
/* Aufgabe 34b*/

#include <stdio.h>

int multiples_of_x(int n, int x, int *lower, int *greater);

int main(void)
{
	int a, b;
	if(multiples_of_x(20, 7,&a, &b)){
		printf("multiples_of_x(20, 7,&a, &b):\n");
		printf("lower: %i\n", a);
		printf("greater: %i\n", b);
	}
	printf("\n");
	if(multiples_of_x(50, 10,&a, &b)){
		printf("multiples_of_x(50, 10,&a, &b)\n");
		printf("lower: %i\n", a);
		printf("greater: %i\n", b);
	}
	printf("\n");
	if(multiples_of_x(2032, 123,&a, &b)){
		printf("multiples_of_x(2032, 123,&a, &b)\n");
		printf("lower: %i\n", a);
		printf("greater: %i\n", b);
	}
	printf("\n");
	if(multiples_of_x(213, 1,&a, &b)){
		printf("multiples_of_x(213, 1,&a, &b)\n");
		printf("lower: %i\n", a);
		printf("greater: %i\n", b);
	}
	printf("\n");
	if(multiples_of_x(2, 1,&a, &b)){
		printf("multiples_of_x(2, 1,&a, &b)\n");
		printf("lower: %i\n", a);
		printf("greater: %i\n", b);
	}printf("\n");
	if(multiples_of_x(0, 0,&a, &b)){
		printf("multiples_of_x(0, 0,&a, &b)\n");
		printf("lower: %i\n", a);
		printf("greater: %i\n", b);
	}

	return 0;
}

int multiples_of_x(int n, int x, int *lower, int *greater)
{
	int i = 1;

	if(lower == NULL || greater == NULL){
		return 1;
	}

	do{
		*lower = (i * x);
		++i;

	}while((i * x) < n);

	i = n;
	do{
		*greater = (i * x);
		--i;
	}while((i * x) > n);

	return 1;
}
```

```c
/* Aufgabe 34c*/
#include <stdio.h>

int read_percentage(int *percentage);
void flush();

int main(void)
{
	int a;
	if(read_percentage(&a)){
		printf("Die Eingabe war erfolgreich: ");
		printf("%i\n", a);
	}else{
		printf("Ungültige Eingabe\n");
	}


	return 0;
}

int read_percentage(int *percentage)
{
	int status;
	printf("Geben Sie eine Zahl zwischen 0 und 100 ein: ");
	status = scanf("%d", percentage);

	if(status == 1 && *percentage >= 0 && *percentage <= 100){
		return 1;
	}else{
		flush();
		return 0;
	}
}

void flush()
{
	while(getchar() != '\n'){

	}
}
```

```c
/* Aufgabe 35a*/

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

char *str_rchr(const char *cs, int c);

int main(void)
{
	printf("%s\n", str_rchr("", 'g'));
	printf("%s\n", str_rchr("", '\0'));
	printf("%s\n", str_rchr("Schokolade", 'o'));
	printf("%s\n", str_rchr("Schokolade", 'x'));


	return 0;
}

char *str_rchr(const char *cs, int c)
{
	char *p = NULL;

	if(cs == NULL && (char)c != '\0'){


		return NULL;
	}

	while(*cs != '\0'){
		if( *cs == (char)c){
			p = (char *)cs;
		}
		++cs;
	}
	return p;
}
```

```
/* Aufgabe 35b*/

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

const char *str_str(const char *cs, const char *ct);
int str_len(const char str[]);

int main(void)
{

	printf("1. %s\n", str_str("Schokolade", ""));
	printf("2. %s\n", str_str(" ", " "));
	printf("3. %s\n", str_str("", ""));
	printf("4. %s\n", str_str("Schokolade", "Info"));
	printf("5. %s\n", str_str("Schokolade", "lade"));


	return 0;
}

const char *str_str(const char *cs, const char *ct)
{
	int i;
	int j;
	int flag;
	int lencs;
	int lenct;

	lencs = str_len(cs);
	lenct = str_len(ct);

	for(i = 0; i < lencs; ++i){
		if(cs[i] == ct[0]){
			flag = 0;

			for(j = 0; j < lenct; ++j){
				if(cs[i + j] != ct[j]){
					flag = 1;
					break;
				}
			}
		}
		if(flag == 0){
			return &cs[i];
		}
	}
	return NULL;

}

int str_len(const char str[])
{
        int count;

        for(count = 0; str[count] != '\0'; ++count);

        return count;
}
```

```c
/* Aufgabe 35c*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <string.h>

int main(void)
{
	int i;
	char token[20];
	char input[20];
	char *hours;
	char *minutes;
	char *seconds;

	printf("Geben Sie eine Uhrzeit ein: ");
	scanf("%s", input);

	strcpy(token, input);

	i = 1;
	strtok(token, ":-/");
	while(strtok(NULL, ":-/") != NULL){
		++i;
	}

	if(i == 3){

		hours = strtok(input, ":-/");
		minutes = strtok(NULL, ":-/");
		seconds = strtok(NULL, ":-/");

		printf("hours: %s, minutes: %s, seconds: %s\n", hours, minutes, seconds);



	}else{
			printf("Formatfehler!\n");

	}

	return 0;

}
```

```
/* Aufgabe 36a*/

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

int *array_d_copy(int v[], int n);

int main(void)
{
	int i;
	int v[] = {1, 2, 3, 4, 5};
	int *d = array_d_copy(v, 5);


	for(i = 0; i < 5; ++i){
		printf("%i\n",d[i]);
	}
	return 0;
}

int *array_d_copy(int v[], int n)
{
	int i;
	int *copy = malloc(n * sizeof(int));

	if(copy == NULL){
		return NULL;
	}

	for(i = 0; i < n; ++i){

		copy[i] = v[i];
	}

	return copy;
	free(copy);
}
```

```c
/* Aufgabe 36b*/

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


int *array_d_shuffle(int v[], int w[], int n);
int main(void)
{
	int i;
	int v[] = {1, 2, 3};
	int w[] = {4, 5, 6};
	int *d = array_d_shuffle(v, w, 3);


	for(i = 0; i < 6; ++i){
		printf("%i\n",d[i]);
	}
	return 0;


}

int *array_d_shuffle(int v[], int w[], int n)
{
	int i;

	int *shuffled_v = malloc((2 * n) * sizeof(int));

	if(shuffled_v == NULL){
		return NULL;
	}

	for(i = 0; i < n; ++i){
		shuffled_v[i * 2] = v[i];
		shuffled_v[i * 2 + 1] = w[i];
	}

	return shuffled_v;
	free(shuffled_v);


}
```

```c
/* Aufgabe 36c*/

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


int *array_d_shuffle(int v[], int w[], int n);
int *array_d_copy(int v[], int n);
int main(void)
{
	int i;
	int array[] = {1, 3, 2, 5, 5};
	int *array_copy = array_d_copy(array, 5);
	int *array_shuffle = array_d_shuffle(array, array_copy, 4);

	if(array_copy == NULL){
		printf("Programmfehler!\n");
	}

	if(array_shuffle == NULL){
		printf("Programmfehler!\n");
	}

	for(i = 0; i < 8; ++i){
		printf("%i\n",array_shuffle[i]);
	}
	return 0;


}

int *array_d_shuffle(int v[], int w[], int n)
{
	int i;

	int *shuffled_v = malloc((2 * n) * sizeof(int));

	if(shuffled_v == NULL){
		return NULL;
	}

	for(i = 0; i < n; ++i){
		shuffled_v[i * 2] = v[i];
		shuffled_v[i * 2 + 1] = w[i];
	}

	return shuffled_v;
	free(shuffled_v);


}

int *array_d_copy(int v[], int n)
```

```
{
	int i;
	int *copy = malloc(n * sizeof(int));

	if(copy == NULL){
		return NULL;
	}

	for(i = 0; i < n; ++i){

		copy[i] = v[i];
	}

	return copy;
	free(copy);
}
```