Aufgabe 26a)

```
#include <stdio.h>

#define INPUT_TOO_LONG -1
#define VALID_INPUT 1
#define NO_BINARY 2
#define DIM 10


int flush_buff (void){
	int c;
	while ((c = getchar ()) != '\n' && c != EOF ){
	}
	return c != EOF;
}

int read_binary(int b[]){
	int i = 0;
	int c = getchar();
	while (c != '\n' && i < DIM -1){
		b[i++] = c;
		c = getchar();
	}
	if (i == DIM -1 && c != '\n'){
		flush_buff();
		return INPUT_TOO_LONG;
	}
	for(i = 0; i < DIM; i++){
		if(b[i] != '0' || b[i] != '1'){
			flush_buff();
			return NO_BINARY;
		}
	}
	b[i] = '\0';
	return VALID_INPUT;
}

int main(){
	int in[DIM];
	printf("Binärcodierung mit %d-Stellen eingeben:\n", DIM);
	if (read_binary(in)== -1){
		printf("Eingabe zu lang\n");
	}
	if (read_binary(in)== 2){
		printf("Eingabe nicht in binaer!\n");
	}
	if (read_binary(in)== 1){
		printf("Eingabe gespeichert\n");
	}
	return 0;
}
```

Aufgabe 26b)

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

#define INPUT_TOO_LONG -1
#define VALID_INPUT 1
#define NO_BINARY 2
#define NAN 3
#define K_TOO_HIGH 4

#define DIM 10

double decode_fk(int festkomma[], int n, int k);
int read_binary(int b[]);
int flush_buff (void);


int main(void)
{
    int k, in[DIM];
    scanf("Anzahl der Nachkommastellen für 10-Bit eingeben\n%d", &k);
    if(!isdigit(k)){
        printf("Eingabe ist keine Zahl\n");
        return NAN;
    }
    if(k>DIM){
        printf("K ist zu hoch gewaehlt!\n");
        return K_TOO_HIGH;
    }
       read_binary(in);
       printf("Der decodierte Wert ist: %f\n", decode_fk(in, DIM, k));
       return 0;
}

double decode_fk(int festkomma[], int n, int k){
       double result;
       int i;
       result = 0;

       for(i = 0; i < n; i++){
              result *= 2;
              result += festkomma[i];
       }
       return result / pow(2, k);
}

int read_binary(int b[]){
    int i = 0;
    int c = getchar();
    while (c != '\n' && i < DIM -1){
        b[i++] = c;
        c = getchar();
    }
    if (i == DIM -1 && c != '\n'){
        flush_buff();
        return INPUT_TOO_LONG;
    }
    for(i = 0; i < DIM; i++){
        if(b[i] != '0' || b[i] != '1'){
            flush_buff();
            return NO_BINARY;
        }
    }
    b[i] = '\0';
    return VALID_INPUT;
}

int flush_buff (void){
    int c;
    while ((c = getchar ()) != '\n' && c != EOF ){
    }
    return c != EOF;
}
```