Aufgabe 13)

| $x$ | $c_{1K,4}(x)$ | $c_{2K,4}(x)$ | $c_{1K,8}(x)$ | $c_{2K,8}(x)$ |
|---|---|---|---|---|
| 3 | 0011 | 0011 | 00000011 | 00000011 |
| 9 | 1001 | 1001 | 00001001 | 0001001 |
| -111 | X | X | 10010000 | 10010001 |
| $-8$ | 0111 | 1000 | 11110111 | 11111000 |
| 7 | 0111 | 0111 | 00000111 | 0000111 |
| 105 | X | X | 01101001 | 01101001 |
| 128 | X | X | 10000000 | 10000000 |

Aufgabe 14)

a)

| | | | | |
|---|---|---|---|---|
| 0101 | $\oplus_{1K,4}$ | 0001 | = | 0110 |
| 0011 | $\ominus_{1K,4}$ | 0111 | = | 1011 |
| 0101 | $\oplus_{2K,4}$ | 0011 | = | 1000 |
| 1000 | $\ominus_{2K,4}$ | 0101 | = | 0010 (overflow) |
| 10010110 | $\oplus_{1K,8}$ | 00110100 | = | 11001110 |
| 01011010 | $\ominus_{1K,8}$ | 00011111 | = | 01110001 |
| 11010010 | $\oplus_{2K,8}$ | 10110001 | = | 01010011 (of) |
| 10000000 | $\ominus_{2K,8}$ | 00000001 | = | 01111110 (of) |

1
0101
0001
===
0110

.

b)

| Rechenausdruck | $c_{1K,8}$ | $c_{2K,8}$ |
|---|---|---|
| $50+77$ = | 127 | 127 |
| $41+97$ = | 138 | 138 |
| $-17-110$ = | -127 | -127 |
| $-95-33$ = | 126 | -128 |

-17-110 =
-00010001
-01101110
======= c1k8
+11101111
+10010001
=======
+10000000 = -127

-17-110 = -17 + (-110) =(c2K8)=
-00010001
-01101110
=======
+11101111
+10010010
=======
+10000001; -01111110 -1 = -127

-95-33=
-01011111
-00100001
====== c1k8
10100000
11011110
======
01111110 = 126

-95-33=
-01011111
-00100001
======= c2k8
10100001
11011111
======
10000000 = -01111111-1 = -128

Aufgabe 15)

1. 1
2. {000,001,010,011,100,101,110,111}
3. 10000
4. e, 01, 10, 000, 001, 0001
5. {e, 1, 0, 01, 11, 001, 0011, 00111, 111, 0111}
6. 20
7. {x|x in B* mit x = ab und b=00}
8. {x|x in D* mit |x|=2}
9. A^0 = {e}
10. {x|x in B* mit |x| > 0 und Quersumme(x) = 0}
11. 00000; 100100, 1000
12. {x|x in D* x = ab; b = 3, a = n-0er, n in N}
13. (0^(512))1, (0*(111))1

Aufgabe 16)

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

#define INT_BIT sizeof(int)*8

void print_bits(int b[], int size)
{
        for (int i = size-1; i >= 0; i--)
        {
                printf("%i", b[i]);
        }
        printf("\n");
}

void init_bits(int b[], int size)
{
        for (int i = 0; i < size; i++)
        {
                b[i] = 0;
        }
}

int pow69(int n)
{
        int p;
        p = 1;
        for (int i = 0; i < n; i++)
        {
                p = p + p;
        }
        return p;
}

void get_bits(int b[], int n)
{
        for (int i = INT_BIT-1; i > -1; i--)
        {
                int p;
                p = pow69(i);
                if (n - p > -1)
                {
                        b[i] = 1;
                        n = n - p;
                        if (n == 0) return;
                }
        }
}
```

```c
/*
Integer Overflow bei -2147483648-1 und 2147483647+1
*/
int get_int(int b[])
{
        int r;
        int flip;
        flip = 0;
        r = 0;
        for (int i = INT_BIT-1; i > -1; i--)
        {

                if (b[i] == 1 && i == INT_BIT-1)
                {
                        flip = 1;
                }
                if (flip == 1)
                {
                        /*
                        Kann man wahrscheinlich besser lösen...
                        */
                        if (b[i] == 1) b[i] = 0;
                        else if (b[i] == 0) b[i] = 1;
                }
        }
        for (int i = INT_BIT-1; i > -1; i--)
        {
                if (b[i] == 1)
                {
                        r += pow69(i);
                }
        }
        if (flip == 1)
        {
                return -r-1;
        }
        return r;
}


int main(int argc, char *argv[])
```

```c
int main(int argc, char *argv[])
{
        int a;
        int b[INT_BIT];
        printf("INT_BIT=%li\n", INT_BIT);
        init_bits(b, INT_BIT);
        if (argc > 1) {
                get_bits(b, atoi(argv[1]));
        } else {
                get_bits(b, 420);
        }
        print_bits(b, INT_BIT);
        a = get_int(b);
        printf("a=%i\n", a);

        return 0;
}
```

```
mkypr@Aspire:~/studium/info1/exercise/04$ gcc aufgabe16.c && ./a.out 2147483647
INT_BIT=32
01111111111111111111111111111111
a=2147483647
```

```
mkypr@Aspire:~/studium/info1/exercise/04$ gcc aufgabe16.c && ./a.out -15
INT_BIT=32
11111111111111111111111111110001
a=-15
```