

Übungsblatt 2

Aufgabe A – KNF & DNF

1. Finden sie die disjunktive Normalform und die konjunktive Normalform zur folgenden aussagenlogischen Formel:

$$(A \vee B) \Rightarrow C)$$

2. Finden sie die disjunktive Normalform und die konjunktive Normalform zur folgenden aussagenlogischen Formel:

$$(A \wedge B) \Leftrightarrow C)$$

3. Finden Sie die disjunktive Normalform und die konjunktive Normalform zur folgenden aussagenlogischen Formel:

$$(A \vee B) \Rightarrow \neg C) \Rightarrow D$$

Aufgabe B – Bit Hacks in C

Folgenden Code soll kompiliert und ausgeführt werden.

```
1 gcc program_name -o compiled_name
```

1. Kompilieren sie das Programm *p1.c* und führen sie dies für natürliche Zahlen aus.
2. Testen sie verschiedene Eingaben. Stellen sie eine erste Vermutung auf!
3. Schauen sie in den Quellcode – nur die `main`-Funktion: Der Ausdruck `f = (v & (v - 1)) == 0;` ist relevant. Der Rest dient der Ein- und Ausgabe.
4. Nehmen sie eine Wert und gehen den logischen Ausdruck Schritt für Schritt durch.

```
1  /*
2  Compilation command for amd_64:
3  gcc -o filename outname
4
5  */
6  #include <stdio.h>
7  #include <stdlib.h>
8  #include <stdbool.h>
9
10 void bin(unsigned n) {
11     unsigned i;
12     for (i = 1 << 31; i > 0; i = i / 2)
13         (n & i) ? printf("1") : printf("0");
```

```
14 }
15
16 int main(int argc, char **argv) {
17     unsigned int v;
18     printf("Input:");
19     scanf("%d", &v);
20     bool f;
21     f = (v & (v - 1)) == 0;
22     printf("Input was: %d\n", v);
23     printf("Binary was :");
24     bin(v);
25     printf("\n");
26     printf("Result: f: %d\n", f);
27 }
```

5. Kompilieren sie das Programm *p2.c* und führen sie dies für natürliche Zahlen aus.
6. Testen sie verschiedene Eingaben. Stellen sie eine erste Vermutung auf!
7. Schauen sie in den Quellcode – nur die *main*-Funktion: Der Ausdruck $r = (n \wedge (n - 1)) * v$; ist relevant. Der Rest dient der Ein- und Ausgabe.

```
1  /*
2  Compilation command for amd_64:
3  gcc -o filename outname
4
5  */
6  #include <stdio.h>
7  #include <stdlib.h>
8  #include <stdbool.h>
9
10 void bin(unsigned n) {
11     unsigned i;
12     for (i = 1 << 31; i > 0; i = i / 2)
13         (n & i) ? printf("1") : printf("0");
14 }
15
16 int main(int argc, char **argv) {
17     bool n;
18     int v;
19     int r;
20
21     printf("No Magic 0/1 (Y/N)? ");
22     scanf("%d", &n);
23     printf("Input to put magic on: ");
24     scanf("%d", &v);
25     r = (n ^ (n - 1)) * v;
26
27     printf("Input was: %d\n", v);
```

```
28     printf("Binary was:\t");
29     bin(v);
30     printf("\n");
31     printf("Result: r:\t", r);
32     bin(r);
33     printf("\n");
34 }
```

8. Nehmen sie eine Wert und gehen den logischen Ausdruck Schritt für Schritt durch.