

Ninja-ASSEMBLER MIT ÜBUNGEN LERNEN

1. ksp 0

Hier brauchen Sie nicht den Ninja-Assembler zu lernen.

2. ksp 1

(VM-Instructionen: halt, pushc, add, sub, mul, div, mod, rdint, wrint, rdchr, wrchr)

A) Teil 1: Ninja_CodeBlock to Ninja-Assembler

```
1 /*Uebungsbeispiel:
2 -1-->auswertung von 2*3+5
3 --->loesung:
4     pushc 2
5     pushc 3
6     mul
7     pushc 5
8     add
9 -2-->The Ninja program fragment:
10     writeInteger((3 + 4) * (10 - 6));
11     writeCharacter('\n');
12 --->Loesung:
13     pushc 3
14     pushc 4
15     add
16     pushc 10
17     pushc 6
18     sub
19     mul
20     wrint
21     pushc 10
22     wrchr
23     halt
24 1) auswertung von 2*(3+5)
25 ----->loesung: ?
26 2) auswertung von (-12 * (3-2) / 2) % 5
27 ----->loesung: ?
28 3) auswertung von ((15 /3) + (2%7))
29 ----->loesung: ?
30 4) auswertung von -((-12 * (3-2) / 2) % 5)
31 ----->loesung: ?
32 5) auswertung von -(1+2+3+4+5+6*4-2)
33 ----->loesung: ?
34 5) auswertung von -(1-2-3-4-5-6%4+2)
35 ----->loesung: ?
36 6)The Ninja program fragment:
37     writeInteger(-2 * readInteger() + 3);
38     writeCharacter('\n');
39 ----->loesung: ?
40 7)The Ninja program fragment:
41     writeInteger(readCharacter());
42     writeCharacter('\n');
43 ----->loesung: ?
44 8)The Ninja program fragment:
45     writeInteger(readCharacter()+12-10);
46     writeCharacter('\n');
47 ----->loesung: ?
```

```
48 9)The Ninja program fragment:
49     writeInteger(198*readInteger()+12);
50     writeCharacter('\n');
51 ----->loesung: ?
52 10)The Ninja program fragment:
53     writeCharacter('a');
54     writeCharacter(':');
55     readInteger();
56     writeCharacter('\n');
57     writeCharacter('b');
58     writeCharacter(':');
59     readInteger();
60     writeCharacter('\n');
61 ----->loesung: ?
62
```

B) Teil 2: Ninja-Assembler to Auswertung

```
1 /*Uebungsbeispiel:
2     pushc 2
3     pushc 3
4     mul
5     pushc 5
6     add
7 -->auswertung: 2*3+5 = 11
8
9 standart eingabe ist 5 für rdint und 'a' für rdchr
10 1)
11     rdint
12     pushc 10
13     add
14 -->auswertung:
15
16 2)
17     rdchr
18     pushc 10
19     add
20 -->auswertung:
21 3)
22     rdint
23     pushc 10
24     pushc 22
25     pushc 44
26     pushc 100
27     add
28     sub
29     mul
30     add
31 -->auswertung:
32 4)
33     pushc 4
34     pushc 3
35     mod
36     pushc 19
37     add
38     pushc 2
39     div
40     rdchr
41     add
42 -->auswertung:
43 5)
44     pushc 1000
45     add
46 -->auswertung:
47
48 */
```

3. ksp 2

(VM-Instruktionen: halt, pushc, add, sub, mul, div, mod, rdint, wrint, rdchr, wrchr, pushg, popg, asf, rsf, pushl, popl)

A- Teil 1: Ninja-CodeBlock to Ninja-Assembler

```
1  /*Beispiel:
2  global Integer x;
3  global Integer y;
4  x = 2;
5  y = x + 3;
6  x = 7 * y + x;
7  writeInteger(x + -33);
8  writeCharacter('\n');
9  -----
10 Antwort:          */
11     pushc    2
12     popg     0
13     pushg    0
14     pushc    3
15     add
16     popg     1
17     pushc    7
18     pushg    1
19     mul
20     pushg    0
21     add
22     popg     0
23     pushg    0
24     pushc    -33
25     add
26     wrint
27     pushc    '\n'
28     wrchr
29     halt
30 /*1) Ninja-Blockcode:
31 global Integer a;
32 a = 2;
33 a = 2*a - 5;
34 writeInteger(x + 10);
35 writeCharacter('\n');
36 -----
37 Antwort:
38 */
```

```
1  /*Beispiel:
2  local Integer x;
3  local Integer y;
4  x = 2;
5  y = x + 3;
6  x = 7 * y + x;
7  writeInteger(x + -33);
8  writeCharacter('\n');
9  -----
10 Antwort:          */
11     asf 2
12     pushc    2
13     popl     0
14     pushl    0
15     pushc    3
16     add
17     popl     1
18     pushc    7
19     pushl    1
20     mul
21     pushl    0
22     add
23     popl     0
24     pushl    0
25     pushc    -33
26     add
27     wrint
28     pushc    '\n'
29     wrchr
30     rsf
31     halt
32 /*2) Ninja-Blockcode:
33 global Integer a;
34 global Integer b;
35 global Integer c;
36 a = readInteger();
37 b = readInteger();
38 c = readInteger();
39 c = a+b+c;
40 writeInteger(2*c);
41 writeCharacter('\n');
42 -----
43 Antwort:
44 */
```

ASM aufgabe5.asm

```

1  /*3) Ninja-Blockcode:
2  global Integer a;
3  a = readInteger();
4  a = (10 % a) + 18;
5  writeInteger(a);
6  writeCharacter('\n');
7  -----
8  Antwort:|
9  */
10 /*4) Ninja-Blockcode:
11 global Integer a;
12 a = readCharacter();
13 a = (a+10);
14 writeCharacter(a);
15 writeCharacter('\n');
16 -----
17 Antwort:
18 */
19 /*5) Ninja-Blockcode:
20 local Integer a;
21 a = 2;
22 a = 2*a - 5;
23 writeInteger(x + 10);
24 writeCharacter('\n');
25 -----
26 Antwort:
27 */
28 /*6) Ninja-Blockcode:
29 local Integer a;
30 local Integer b;
31 local Integer c;
32 a = readInteger();
33 b = readInteger();
34 c = readInteger();
35 c = a+b+c;
36 writeInteger(2*c);
37 writeCharacter('\n');
38 -----
39 Antwort:
40 */

```

```

1  /*3) Ninja-Blockcode:
2  global Integer a;
3  a = readInteger();
4  a = (10 % a) + 18;
5  writeInteger(a);
6  writeCharacter('\n');
7  -----
8  Antwort:
9  */
10 /*4) Ninja-Blockcode:
11 global Integer a;
12 a = readCharacter();
13 a = (a+10);
14 writeCharacter(a);
15 writeCharacter('\n');
16 -----
17 Antwort:
18 */
19 /*5) Ninja-Blockcode:
20 local Integer a;
21 a = 2;
22 a = 2*a - 5;
23 writeInteger(x + 10);
24 writeCharacter('\n');
25 -----
26 Antwort:
27 */
28 /*6) Ninja-Blockcode:
29 local Integer a;
30 local Integer b;
31 local Integer c;
32 a = readInteger();
33 b = readInteger();
34 c = readInteger();
35 c = a+b+c;
36 writeInteger(2*c);
37 writeCharacter('\n');
38 -----
39 Antwort:
40 */

```

ASM aufgabe6.asm

```

1  /*7) Ninja-Blockcode:
2  local Integer a;
3  a = readInteger();
4  a = (10 % a) + 18;
5  writeInteger(a);
6  writeCharacter('\n');
7  -----
8  Antwort:
9  */
10 /*8) Ninja-Blockcode:
11 local Integer a;
12 a = readCharacter();
13 a = (a+10);
14 writeCharacter(a);
15 writeCharacter('\n');
16 -----
17 Antwort:
18 */
19 /*9) Ninja-Blockcode:
20 global Integer a;
21 a = 19;
22 ooooooooooooooooooooooooooooooooooooooooooooo
23 local Character b;
24 b = 'a';
25 writeCharacter(b);
26 writeCharacter('\n');
27 -----
28 Antwort:
29 */
30 /*10) Ninja-Blockcode:
31 global Integer a;
32 global Integer b;
33 a = 19;
34 b = -100;
35 ooooooooooooooooooooooooooooooooooooooooooooo
36 local Integer x;
37 x = b / a;
38 writeInteger(x);
39 writeCharacter('\n');
40 -----
41 Antwort:
42 */

```

```

1  /*7) Ninja-Blockcode:
2  local Integer a;
3  a = readInteger();
4  a = (10 % a) + 18;
5  writeInteger(a);
6  writeCharacter('\n');
7  -----
8  Antwort:
9  */
10 /*8) Ninja-Blockcode:
11 local Integer a;
12 a = readCharacter();
13 a = (a+10);
14 writeCharacter(a);
15 writeCharacter('\n');
16 -----
17 Antwort:
18 */
19 /*9) Ninja-Blockcode:
20 global Integer a;
21 a = 19;
22 ooooooooooooooooooooooooooooooooooooooooooooo
23 local Character b;
24 b = 'a';
25 writeCharacter(b);
26 writeCharacter('\n');
27 -----
28 Antwort:
29 */
30 /*10) Ninja-Blockcode:
31 global Integer a;
32 global Integer b;
33 a = 19;
34 b = -100;
35 ooooooooooooooooooooooooooooooooooooooooooooo
36 local Integer x;
37 x = b / a;
38 writeInteger(x);
39 writeCharacter('\n');
40 -----
41 Antwort:
42 */

```

B- Teil 2: Ninja-Assembler to c-CodeBlock

4. ksp 3

(VM-Instruktionen: halt, pushc, add, sub, mul, div, mod, rdint, wrint, rdchr, wrchr pushg, popg, asf, rsf, pushl, popl, eq, ne, lt, le, gt, ge, jmp, brf, brt)

A- Teil 1: Ninja-CodeBlock to Ninja-Assembler

```
1  /*Beispiel:
2  global Integer a;
3  a = readInteger();
4  if(a<0){
5      writeInteger(0);
6  }else{
7      writeInteger(1);
8  }
9  writeCharacter('\n');
10 -----
11 Antwort:          */
12     rdint
13     popg    0
14     //if(a<0)
15     pushg 0
16     pushc 0
17     lt
18     brf _L1
19     //0
20     pushc 0
21     wrint
22     jmp _L3
23 _L1:
24     //1
25     pushc 1
26     wrint
27     jmp _L3
28
29 _L3:
30     pushc 10
31     wrchr
32     halt
33 /*1) Ninja-Blockcode:
34 local Integer a;
35 a = readInteger();
36 if(a<0){
37     writeInteger(0);
38 }else{
39     writeInteger(1);
40 }
41 writeCharacter('\n');
42 -----
43 Antwort:
44 */
```

```
1  /*2) Ninja-Blockcode:
2  global Integer a;
3  a = readInteger();
4  if((a%2)==0){
5      writeInteger(1);
6  }else{
7      writeInteger(0);
8  }
9  writeCharacter('\n');
10 -----
11 Antwort:
12 */
13
14 /*3) Ninja-Blockcode:
15 local Integer a;
16 a = readInteger();
17 if((a%2)==0){
18     writeInteger(1);
19 }else{
20     writeInteger(0);
21 }
22 writeCharacter('\n');
23 -----
24 Antwort:
25 */
26
27 /*4) Ninja-Blockcode:
28 global Integer a;
29 global Integer b;
30 a = readInteger();
31 a = readInteger();
32 if(a>b){
33     writeInteger(a);
34 }else{
35     writeInteger(b);
36 }
37 writeCharacter('\n');
38 -----
39 Antwort:
40 */
```


ASM aufgabe9.asm

```

1  /*5) Ninja-Blockcode:
2  local Integer a;
3  local Integer b;
4  a = readInteger();
5  a = readInteger();
6  if(a<b){
7      writeInteger(a);
8  }else{
9      writeInteger(b);
10 }
11 writeCharacter('\n');
12 -----
13 Antwort:
14 */
15 /*6) Ninja-Blockcode:
16 global Integer a;
17 a = readInteger();
18 while(a>0){
19     writeCharacter('a');
20     writeCharacter(':');
21     writeInteger(a);
22     writeCharacter('\n');
23     a = a - 1;
24 }
25 writeCharacter('\n');
26 -----
27 Antwort:
28 */
29 /*7) Ninja-Blockcode:
30 local Integer a;
31 a = readInteger();
32 while(a>=0){
33     writeCharacter('a');
34     writeCharacter(':');
35     writeInteger(a);
36     writeCharacter('\n');
37     a = a - 1;
38 }
39 writeCharacter('\n');
40 -----
41 Antwort:
42 */
43

```

```

1  /*5) Ninja-Blockcode:
2  local Integer a;
3  local Integer b;
4  a = readInteger();
5  a = readInteger();
6  if(a<b){
7      writeInteger(a);
8  }else{
9      writeInteger(b);
10 }
11 writeCharacter('\n');
12 -----
13 Antwort:
14 */
15 /*6) Ninja-Blockcode:
16 global Integer a;
17 a = readInteger();
18 while(a>0){
19     writeCharacter('a');
20     writeCharacter(':');
21     writeInteger(a);
22     writeCharacter('\n');
23     a = a - 1;
24 }
25 writeCharacter('\n');
26 -----
27 Antwort:
28 */
29 /*7) Ninja-Blockcode:
30 local Integer a;
31 a = readInteger();
32 while(a>=0){
33     writeCharacter('a');
34     writeCharacter(':');
35     writeInteger(a);
36     writeCharacter('\n');
37     a = a - 1;
38 }
39 writeCharacter('\n');
40 -----
41 Antwort:
42 */
43

```

ASM aufgabe10.asm

```

1  /*8) Ninja-Blockcode:
2  local Integer a;
3  local Integer resultat;
4  a = readInteger();
5  resultat = 1;
6  writeInteger(a);
7  writeCharacter('!');
8  writeCharacter('=');
9  while(a>0){
10     resultat = resultat * a;
11     a = a - 1;
12 }
13 writeInteger(resultat);
14 writeCharacter('\n');
15 -----
16 Antwort:
17 */
18 /*9) Schreiben Sie ein Programm
19 in Ninja-Assembler, das prüft,
20 ob die eingegebene Zahl a
21 eine primzahl ist:
22 -->gibt das Programm 1 aus, folgt
23 a ist eine primzahl.
24 -->gibt das Programm 0 aus, folgt
25 a ist keine primzahl.
26 -----
27 Antwort:
28 */
29 /*10) Ninja-Blockcode:
30 Schreiben Sie ein Programm
31 in Ninja-Assembler, das die
32 Fibonnaci eine eingegebene Zahl a
33 rechnen.
34 -----
35 Antwort:
36 */

```

```

1  /*8) Ninja-Blockcode:
2  local Integer a;
3  local Integer resultat;
4  a = readInteger();
5  resultat = 1;
6  writeInteger(a);
7  writeCharacter('!');
8  writeCharacter('=');
9  while(a>0){
10     resultat = resultat * a;
11     a = a - 1;
12 }
13 writeInteger(resultat);
14 writeCharacter('\n');
15 -----
16 Antwort:
17 */
18 /*9) Schreiben Sie ein Programm
19 in Ninja-Assembler, das prüft,
20 ob die eingegebene Zahl a
21 eine primzahl ist:
22 -->gibt das Programm 1 aus, folgt
23 a ist eine primzahl.
24 -->gibt das Programm 0 aus, folgt
25 a ist keine primzahl.
26 -----
27 Antwort:
28 */
29 /*10) Ninja-Blockcode:
30 Schreiben Sie ein Programm
31 in Ninja-Assembler, das die
32 Fibonnaci eine eingegebene Zahl a
33 rechnen.
34 -----
35 Antwort:
36 */

```

B- Teil 2: Ninja-Assembler to c-CodeBlock

5. ksp 4, 5, 6

(VM-Instructionen: halt, pushc, add, sub, mul, div, mod, rdint, wrint, rdchr, wrchr pushg, popg, asf, rsf, pushl, popl, eq, ne, lt, le, gt, ge, jmp, brf, brt, call, ret, drop, pushr, popr, dup)

A- Teil 1: c-CodeBlock to Ninja-Assembler

B- Teil 2: Ninja-Assembler to c-CodeBlock

6. ksp 7, 8

(VM-Instructionen: halt, pushc, add, sub, mul, div, mod, rdint, wrint, rdchr, wrchr pushg, popg, asf, rsf, pushl, popl, eq, ne, lt, le, gt, ge, jmp, brf, brt, call, ret, drop, pushr, popr, dup, new, getf, putf, newa, getfa, getsz, pushn, refeq, refne)

A- Teil 1: c-CodeBlock to Ninja-Assembler

B- Teil 2: Ninja-Assembler to c-CodeBlock