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

```
/*Uebungsbeispiel:
-1-->auswertung von 2*3+5
--->loesung:
        pushc 2
        pushc 3
        mul
        pushc 5
        add
-2-->The Ninja program fragment:
        writeInteger((3 + 4) * (10 - 6));
        writeCharacter('\n');
--->Loesung:
        pushc   3
        pushc   4
        add
        pushc   10
        pushc   6
        sub
        mul
        wrint
        pushc   10
        wrchr
        halt
1) auswertung von 2*(3+5)
----->loesung: ?
2) auswertung von (-12 * (3-2) / 2) % 5
----->loesung: ?
3) auswertung von ((15 /3) + (2%7))
----->loesung: ?
4) auswertung von -((-12 * (3-2) / 2) % 5)
----->loesung: ?
5) auswertung von -(1+2+3+4+5+6*4-2)
----->loesung: ?
5) auswertung von -(1-2-3-4-5-6%4+2)
----->loesung: ?
6)The Ninja program fragment:
         writeInteger(-2 * readInteger() + 3);
         writeCharacter('\n');
----->loesung: ?
7)The Ninja program fragment:
        writeInteger(readCharacter());
        writeCharacter('\n');
----->loesung: ?
8)The Ninja program fragment:
        writeInteger(readCharacter()+12-10);
        writeCharacter('\n');
----->loesung: ?
```

```
9)The Ninja program fragment:
        writeInteger(198*readInteger()+12);
        writeCharacter('\n');
----->loesung: ?
10)The Ninja program fragment:
        writeCharacter('a');
        writeCharacter(':');
        readInteger();
        writeCharacter('\n');
        writeCharacter('b');
        writeCharacter(':');
        readInteger();
        writeCharacter('\n');
----->loesung: ?

```

B) Teil 2: Ninja-Assembler to Auswertung

```
/*Uebungsbeispiel:
        pushc 2
        pushc 3
        mul
        pushc 5
        add
-->auswertung: 2*3+5 = 11

standart eingabe ist 5 für rdint und 'a' für rdchr
1)
        rdint
        pushc 10
        add
-->auswertung:

2)
        rdchr
        pushc 10
        add
-->auswertung:
3)
        rdint
        pushc 10
        pushc 22
        pushc 44
        pushc 100
        add
        sub
        mul
        add
-->auswertung:
4)
        pushc 4
        pushc 3
        mod
        pushc 19
        add
        pushc 2
        div
        rdchr
        add
-->auswertung:
5)
        pushc 1000
        add
-->auswertung:

*/
```

3. ksp 2

(VM-Instructionen: 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

```
/*Beispiel:
global Integer x;
global Integer y;
x = 2;
y = x + 3;
x = 7 * y + x;
writeInteger(x + -33);
writeCharacter('\n');
------------------
Antwort:            */
    pushc   2
    popg    0
    pushg   0
    pushc   3
    add
    popg    1
    pushc   7
    pushg   1
    mul
    pushg   0
    add
    popg    0
    pushg   0
    pushc   -33
    add
    wrint
    pushc   '\n'
    wrchr
    halt
/*1) Ninja-Blockcode:
global Integer a;
a = 2;
a = 2*a - 5;
writeInteger(x + 10);
writeCharacter('\n');
------------------
Antwort:
*/
```

```
/*Beispiel:
local Integer x;
local Integer y;
x = 2;
y = x + 3;
x = 7 * y + x;
writeInteger(x + -33);
writeCharacter('\n');
-----------------------
Antwort:          */
    asf 2
    pushc   2
    popl    0
    pushl   0
    pushc   3
    add
    popl    1
    pushc   7
    pushl   1
    mul
    pushl   0
    add
    popl    0
    pushl   0
    pushc   -33
    add
    wrint
    pushc   '\n'
    wrchr
    rsf
    halt
/*2) Ninja-Blockcode:
global Integer a;
global Integer b;
global Integer c;
a = readInteger();
b = readInteger();
c = readInteger();
c = a+b+c;
writeInteger(2*c);
writeCharacter('\n');
-----------------------
Antwort:
*/
```

aufgabe5.asm

```
/*3) Ninja-Blockcode:
global Integer a;
a = readInteger();
a = (10 % a) + 18;
writeInteger(a);
writeCharacter('\n');
-----------------------
Antwort:
*/
/*4) Ninja-Blockcode:
global Integer a;
a = readCharacter();
a = (a+10);
writeCharacter(a);
writeCharacter('\n');
-----------------------
Antwort:
*/
/*5) Ninja-Blockcode:
local Integer a;
a = 2;
a = 2*a - 5;
writeInteger(x + 10);
writeCharacter('\n');
-----------------------
Antwort:
*/
/*6) Ninja-Blockcode:
local Integer a;
local Integer b;
local Integer c;
a = readInteger();
b = readInteger();
c = readInteger();
c = a+b+c;
writeInteger(2*c);
writeCharacter('\n');
-----------------------
Antwort:
*/
```

aufgabe6.asm

```
/*7) Ninja-Blockcode:
local Integer a;
a = readInteger();
a = (10 % a) + 18;
writeInteger(a);
writeCharacter('\n');
-----------------------
Antwort:
*/
/*8) Ninja-Blockcode:
local Integer a;
a = readCharacter();
a = (a+10);
writeCharacter(a);
writeCharacter('\n');
-----------------------
Antwort:
*/
/*9) Ninja-Blockcode:
global Integer a;
a = 19;
ooooooooooooooooooooooo00000oooooo
local Character b;
b = 'a';
writeCharacter(b);
writeCharacter('\n');
-----------------------
Antwort:
*/
/*10) Ninja-Blockcode:
global Integer a;
global Integer b;
a = 19;
b = -100;
ooooooooooooooooooooooo00000oooooo
local Integer x;
x = b / a;
writeInteger(x);
writeCharacter('\n');
-----------------------
Antwort:
*/
```

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

4. ksp 3

(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)

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

```
/*Beispiel:
global Integer a;
a = readInteger();
if(a<0){
    writeInteger(0);
}else{
    writeInteger(1);
}
writeCharacter('\n');
------------------
Antwort:            */
    rdint
    popg    0
    //if(a<0)
    pushg 0
    pushc 0
    lt
    brf _L1
    //0
    pushc 0
    wrint
    jmp _L3
_L1:
    //1
    pushc 1
    wrint
    jmp _L3

_L3:
    pushc 10
    wrchr
    halt
/*1) Ninja-Blockcode:
local Integer a;
a = readInteger();
if(a<0){
    writeInteger(0);
}else{
    writeInteger(1);
}
writeCharacter('\n');
------------------
Antwort:
*/
```

```

/*2) Ninja-Blockcode:
global Integer a;
a = readInteger();
if((a%2)==0){
    writeInteger(1);
}else{
    writeInteger(0);
}
writeCharacter('\n');
----------------------
Antwort:
*/
/*3) Ninja-Blockcode:
local Integer a;
a = readInteger();
if((a%2)==0){
    writeInteger(1);
}else{
    writeInteger(0);
}
writeCharacter('\n');
----------------------
Antwort:
*/
/*4) Ninja-Blockcode:
global Integer a;
global Integer b;
a = readInteger();
a = readInteger();
if(a>b){
    writeInteger(a);
}else{
    writeInteger(b);
}
writeCharacter('\n');
----------------------
Antwort:
*/
```

aufgabe9.asm

```
/*5) Ninja-Blockcode:
local Integer a;
local Integer b;
a = readInteger();
a = readInteger();
if(a<b){
    writeInteger(a);
}else{
    writeInteger(b);
}
writeCharacter('\n');
----------------------
Antwort:
*/
/*6) Ninja-Blockcode:
global Integer a;
a = readInteger();
while(a>0){
    writeCharacter('a');
    writeCharacter(':');
    writeInteger(a);
    writeCharacter('\n');
    a = a - 1;
}
writeCharacter('\n');
----------------------
Antwort:
*/
/*7) Ninja-Blockcode:
local Integer a;
a = readInteger();
while(a>=0){
    writeCharacter('a');
    writeCharacter(':');
    writeInteger(a);
    writeCharacter('\n');
    a = a - 1;
}
writeCharacter('\n');
----------------------
Antwort:
*/

```

aufgabe10.asm

```
/*8) Ninja-Blockcode:
local Integer a;
local Integer resultat;
a = readInteger();
resultat = 1;
writeInteger(a);
writeCharacter('!');
writeCharacter('=');
while(a>0){
    resultat = resultat * a;
    a = a - 1;
}
writeInteger(resultat);
writeCharacter('\n');
----------------------
Antwort:
*/
/*9) Schreiben Sie ein Programm
in Ninja-Assembler, das prüft,
ob die eingegebene Zahl a
eine  primzahl ist:
-->gibt das Programm 1 aus, folgt
a ist eine primzahl.
-->gibt das Programm 0 aus, folgt
a ist keine primzahl.
----------------------
Antwort:
*/
/*10) Ninja-Blockcode:
Schreiben Sie ein Programm
in Ninja-Assembler, das die
Fibonnaci eine eingegebene Zahl a
rechnen.
----------------------
Antwort:
*/
```

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