

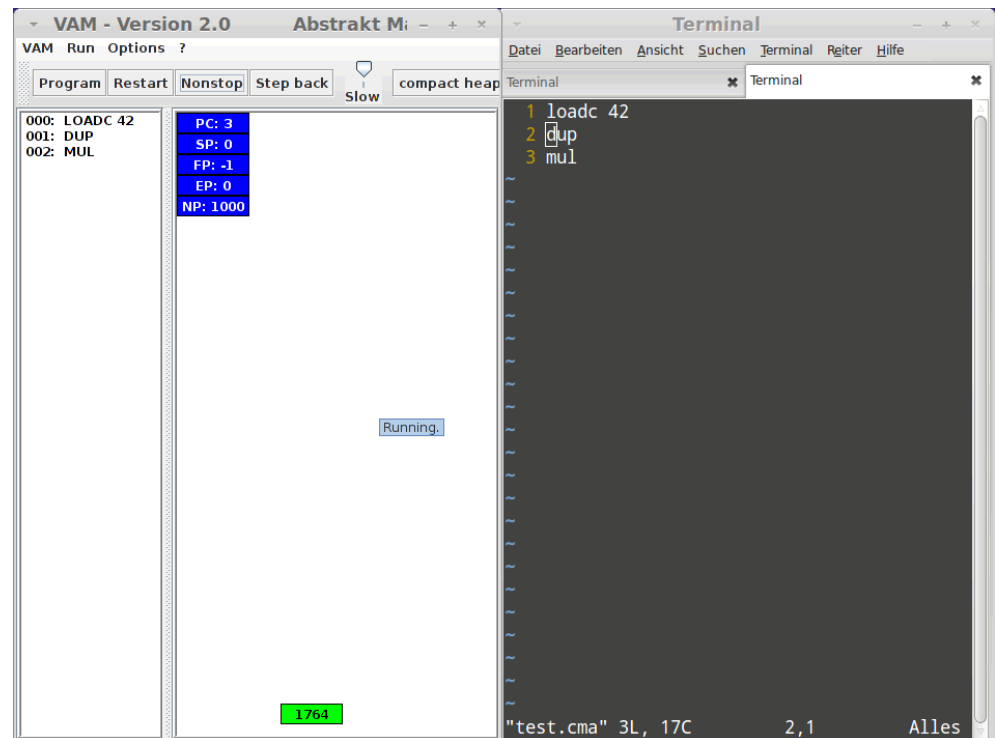
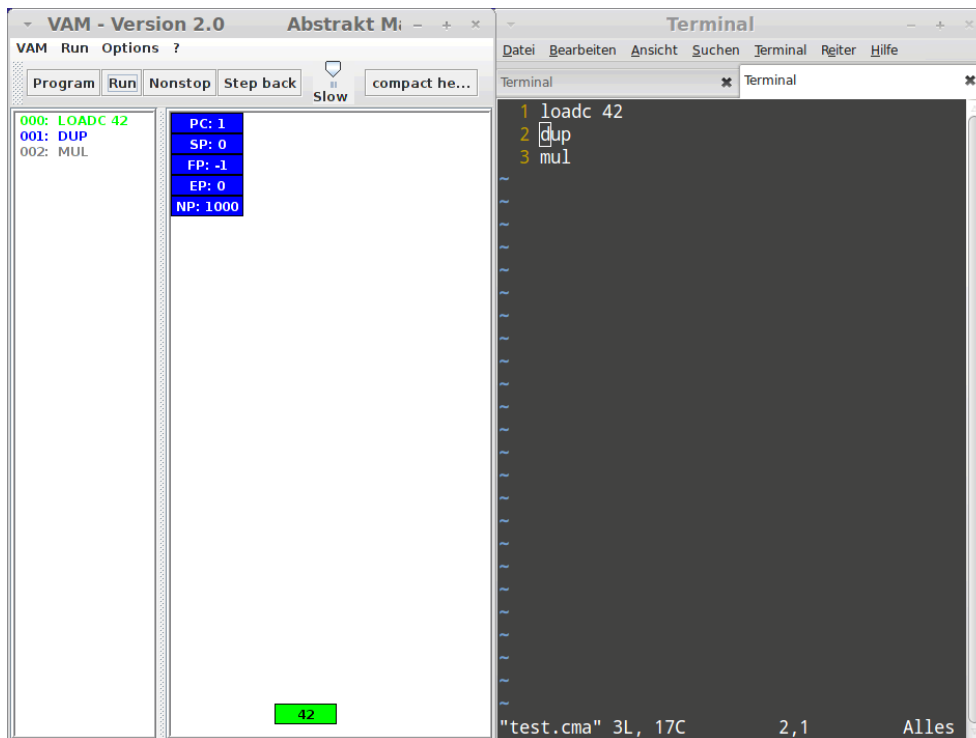
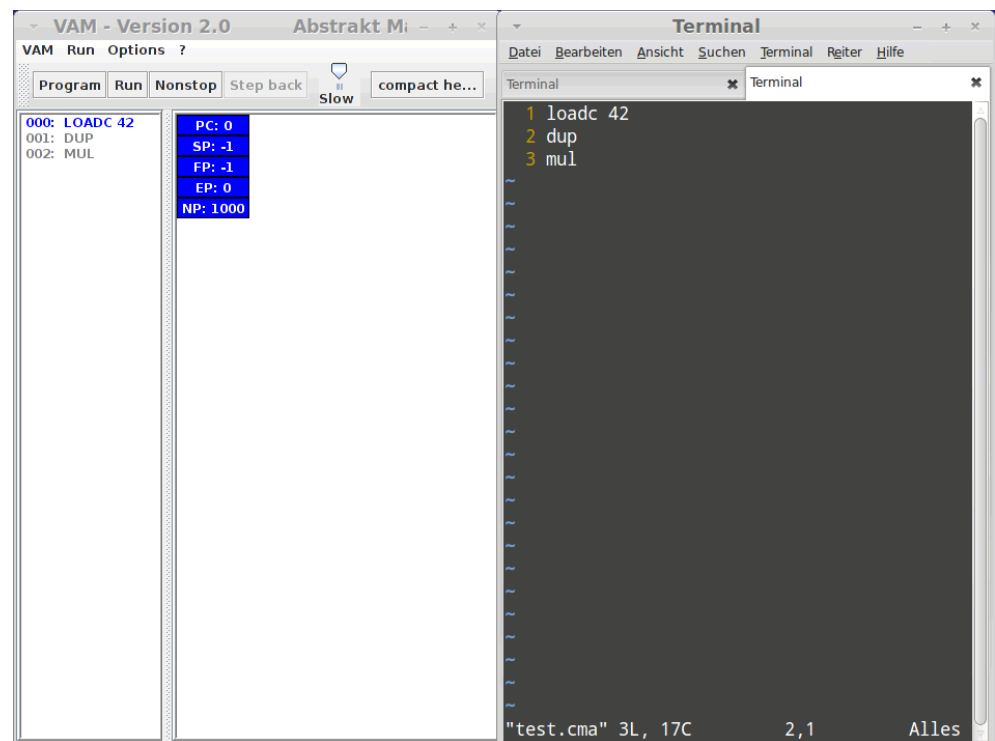
Script generated by TTT

Title: Seidl: Virtual Machines (15.04.2014)

Date: Tue Apr 15 10:16:52 CEST 2014

Duration: 87:14 min

Pages: 40



VAM - Version 2.0 Abstrakt M: - + x

VAM Run Options ?

Program Restart Nonstop Step back Slow compact heap

000: LOADC 42
001: DUP
002: MUL

PC: 3
SP: 0
FP: -1
EP: 0
NP: 1000

1764

Terminal

```

Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal
ttt@tabletseid16 ~/Arbeitsfläche/vam $ ls
dependencies.txt test.cma vam.jar
README.TXT      vam.bat  vam.sh
ttt@tabletseid16 ~/Arbeitsfläche/vam $ cat te
t.cma
loadc 42
dup
mul
ttt@tabletseid16 ~/Arbeitsfläche/vam $ vim ex
.cma
ttt@tabletseid16 ~/Arbeitsfläche/vam $ vim te
st.cma
ttt@tabletseid16 ~/Arbeitsfläche/vam $

```

VAM - Version 2.0 Abstrakt M: - + x

VAM Run Options ?

Program Restart Nonstop Step back Slow compact heap

PC: 0
SP: -1
FP: -1
EP: 0
NP: 1000

Terminal

```

Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal
1 // odd : int -> bool
2
3 // even : int -> bool
4
5 // gcd : int -> int -> int
6 // e.g. gcd(12, 8) = 4

```

"ex.cma" 6L, 95C 1,1 Alles

VAM - Version 2.0 Abstrakt M: - + x

VAM Run Options ?

Program Pause Nonstop Step back Slow compact heap

PC: 0
SP: -1
FP: -1
EP: 0
NP: 1000

Terminal

```

Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal

```

Desktop background: blue with folders (Folien, Persönlicher Ordner) and icons (Adobe Reader, ttt).

Select Program

Pfad- oder Ordnernamen eingeben:
/home/ttt/Arbeitsfläche/vam/

Filter: *

Ordner	Dateien
.	ex.cma.swp .vam_prefs README.TXT dependencies.txt ex.cma test.cma vam.bat vam.jar vam.sh

Filternamen eingeben:
ex.cma

OK Filter Abbrechen

Terminal

```

Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal
1 // odd : int -> bool
2 loadc 8
3 loadc 2
4 mod
5
6 // even : int -> bool
7
8
9 // gcd : int -> int -> int
10 // e.g. gcd(12, 8) = 4
11 // gcd(a, 0) = a
12 // gcd(a, b) = gcd(b, a mod b)
13
14
15 // factorial : int -> int
16
17
18 // strlen : int* -> int
19 // \0-terminated array
20
21
22 // allocate : x -> x*int
23

```

<ma" 23L, 269C geschrieben 2,7 Alles

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact heap

000: LOADC 8
001: LOADC 2
002: MOD

PC: 3
SP: 0
FP: -1
EP: 0
NP: 1000

1 // odd : int -> bool
2 loadc 8
3 loadc 2
4 mod
5
6
7 // even : int -> bool
8 loadc 8
9 loadc 2
10 mod
11 not
12
13
14 // gcd : int -> int -> int
15 // e.g. gcd(12, 8) = 4
16 // gcd(a, 0) = a
17 // gcd(a, b) = gcd(b, a mod b)
18
19
20 // factorial : int -> int
21
22
23 // strlen : int* -> int
24 // \0-terminated array
25
26
27 // allocate : x -> x*int
-- EINFÜGEN -- 5,1 Anfang

0

Select Program

Pfad- oder Ordernamen eingeben:
/home/ttt/Arbeitsfläche/Vam/

Filter

Ordner	Dateien
.	ex.cma.swp .vam_prefs README.TXT dependencies.txt ex.cma test.cma vam.bat vam.jar vam.sh

Filternamen eingeben:
ex.cma

OK Filter Abbrechen

Terminal

1 // odd : int -> bool
2 loadc 3
3 loadc 2
4 mod
5
6 pop
7
8 // even : int -> bool
9 loadc 4
10 loadc 2
11 mod
12 not
13
14 pop
15
16 // gcd : int -> int -> int
17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20
21
22 // factorial : int -> int
23
24
25 // strlen : int* -> int
26 // \0-terminated array
27
<ma" 30L, 302C geschrieben 9,7 Anfang

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact heap

000: LOADC 3
001: LOADC 2
002: MOD
003: POP
004: LOADC 4
005: LOADC 2
006: MOD
007: NOT
008: POP

PC: 9
SP: -1
FP: -1
EP: 0
NP: 1000

5
6 pop
7
8 // even : int -> bool
9 loadc 4
10 loadc 2
11 mod
12 not
13
14 pop
15
16 // gcd : int -> int -> int
17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22
23
24 // factorial : int -> int
25
26
27 // strlen : int* -> int
28 // \0-terminated array
29
30
31 // allocate : x -> x*int
21,9 80%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact heap

000: LOADC 3
001: LOADC 2
002: MOD
003: POP
004: LOADC 4
005: LOADC 2
006: MOD
007: NOT
008: POP

PC: 9
SP: -1
FP: -1
EP: 0
NP: 1000

1 // odd : int -> bool
2 loadc 3
3 loadc 2
4 mod
5
6 pop
7
8 // even : int -> bool
9 loadc 4
10 loadc 2
11 mod
12 not
13
14 pop
15
16 // gcd : int -> int -> int
17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22
23
24 // factorial : int -> int
25 // 0! = 1, 4! = 24, ...
26
27
1,20 Anfang

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

000: LOADC 3
001: LOADC 2
002: MOD
003: POP
004: LOADC 4
005: LOADC 2
006: MOD
007: NOT
008: POP

```

PC: 9
SP: -1
FP: -1
EP: 0
NP: 1000

```

3 //loadc 2
4 //mod
5 //
6 //pop
7 //
8 //// even : int -> bool
9 //loadc 4
10 //loadc 2
11 //mod
12 //not
13 //
14 //pop
15
16 // gcd : int -> int -> int
17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22
23
24 // factorial : int -> int
25 // 0! = 1, 4! = 24, ...
26
27
28 // strlen : int* -> int
29 // \0-terminated array

```

22,0-1 33%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

000: LOADC 3
001: LOADC 2
002: MOD
003: POP
004: LOADC 4
005: LOADC 2
006: MOD
007: NOT
008: POP

```

PC: 9
SP: -1
FP: -1
EP: 0
NP: 1000

```

ttt@tabletseidl6 ~/Arbeitsfläche/vam $

```

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

000: LOADC 3
001: LOADC 2
002: MOD
003: POP
004: LOADC 4
005: LOADC 2
006: MOD
007: NOT
008: POP

```

PC: 9
SP: -1
FP: -1
EP: 0
NP: 1000

```

7 //
8 //// even : int -> bool
9 //loadc 4
10 //loadc 2
11 //mod
12 //not
13 //
14 //pop
15
16 // gcd : int -> int -> int
17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22
23
24 // factorial : int -> int
25 // 0! = 1, 4! = 24, ...
26
27
28 // strlen : int* -> int
29 // \0-terminated array
30
31
32 // allocate : x -> x*int
33

```

29,1 Ende

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

000: LOADC 3
001: LOADC 2
002: MOD
003: POP
004: LOADC 4
005: LOADC 2
006: MOD
007: NOT
008: POP

```

PC: 9
SP: -1
FP: -1
EP: 0
NP: 1000

```

18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22
23
24 // factorial : int -> int
25 // 0! = 1, 4! = 24, ...
26
27
28 // strlen : int* -> int
29 // \0-terminated array
30 loadc 102 // f
31 loadc 111 // o
32 Jump loop
33
34
35 // allocate : x -> x*int
36

```

32,1 Ende

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact heap

```

000: LOADC 3
001: LOADC 2
002: MOD
003: POP
004: LOADC 4
005: LOADC 2
006: MOD
007: NOT
008: POP

```

PC: 9
SP: -1
FP: -1
EP: 0
NP: 1000

```

5 //
6 //pop
7 //
8 //// even : int -> bool
9 //loadc 4
10 //loadc 2
11 //mod
12 //not
13 //
14 //pop
15
16 // gcd : int -> int -> int
17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22
23
24 // factorial : int -> int
25 // 0! = 1, 4! = 24, ...
26
27
28 // strlen : int* -> int
29 // \0-terminated array
30 loadc 102 // f
31 loadc 111 // o

```

5,2 44%

Select Program

Pfad- oder Ordernamen eingeben:

/home/ttt/Arbeitsfläche/Vam/

Filter:

Ordner	Dateien
.	ex.cma.swp .vam_prefs README.TXT dependencies.txt ex.cma test.cma vam.bat vam.jar vam.sh

Filternamen eingeben: ex.cma

OK Filter Abbrechen

```

11 //mod
12 //not
13 //
14 //pop
15
16 // gcd : int -> int -> int
17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22 loop:loada 1
23 jumpz done
24 loada 1
25 loada 0
26 loada 1
27 mod
28 storea 1
29 pop
30 storea 0
31 pop
32 jump loop
33 done: pop
34
35 // factorial : int -> int
36 // 0! = 1, 4! = 24, ...
37
<ma" 47L, 550C geschrieben 22,6

```

50%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 12
001: LOADC 8
002: LOADA 1
003: JUMPZ 13
004: LOADA 1
005: LOADA 0
006: LOADA 1
007: MOD
008: STOREA 1
009: POP
010: STOREA 0
011: POP
012: JUMP 2
013: POP
014: LOADC 102
015: LOADC 111
016: LOADC 111
017: LOADC 0

```

PC: 14
SP: 0
FP: -1
EP: 0
NP: 1000

```

21 loadc 8 // b at 1
22 loop: loada 1
23 jumpz done
24 loada 1
25 loada 0
26 loada 1
27 mod
28 storea 1
29 pop
30 storea 0
31 pop
32 jump loop
33 done: pop
34
35 // factorial : int -> int
36 // 0! = 1, 4! = 24, ...
37
38
39 // strlen : int* -> int
40 // \0-terminated array
41 loadc 102 // f
42 loadc 111 // o
43 loadc 111 // o
44 loadc 0 // \0
45
46 // allocate : x -> x*int
47

```

4 47,0-1 Ende

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 12
001: LOADC 8
002: LOADA 1
003: JUMPZ 13
004: LOADA 1
005: LOADA 0
006: LOADA 1
007: MOD
008: STOREA 1
009: POP
010: STOREA 0
011: POP
012: JUMP 2
013: POP
014: LOADC 102
015: LOADC 111
016: LOADC 111
017: LOADC 0

```

PC: 14
SP: 0
FP: -1
EP: 0
NP: 1000

```

17 // e.g. gcd(12, 8) = 4
18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22 loop: loada 1
23 jumpz done
24 loada 1
25 loada 0
26 loada 1
27 mod
28 storea 1
29 pop
30 storea 0
31 pop
32 jump loop
33 done: pop
34
35 pop
36
37 // factorial : int -> int
38 // 0! = 1, 4! = 24, ...
39
40
41 // strlen : int* -> int
42 // \0-terminated array
43 loadc 102 // f

```

4 38,1 72%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 12
001: LOADC 8
002: LOADA 1
003: JUMPZ 13
004: LOADA 1
005: LOADA 0
006: LOADA 1
007: MOD
008: STOREA 1
009: POP
010: STOREA 0
011: POP
012: JUMP 2
013: POP
014: LOADC 102
015: LOADC 111
016: LOADC 111
017: LOADC 0
  
```

PC: 14
SP: 0
FP: -1
EP: 0
NP: 1000

```

19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22 loop: loada 1
23 jumpz done
24 loada 1
25 loada 0
26 loada 1
27 mod
28 storea 1
29 pop
30 storea 0
31 pop
32 jump loop
33 done: pop
34
35 pop
36
37 // factorial : int -> int
38 // 0! = 1, 4! = 24, ...
39
40
41 // strlen : int* -> int
42 // \0-terminated array
43 loadc 102 // f
44 loadc 111 // o
45 loadc 111 // o
  
```

4

19,1 81%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 12
001: LOADC 8
002: LOADA 1
003: JUMPZ 13
004: LOADA 1
005: LOADA 0
006: LOADA 1
007: MOD
008: STOREA 1
009: POP
010: STOREA 0
011: POP
012: JUMP 2
013: POP
014: LOADC 102
015: LOADC 111
016: LOADC 111
017: LOADC 0
  
```

PC: 14
SP: 0
FP: -1
EP: 0
NP: 1000

```

23 jumpz done
24 loada 1
25 loada 0
26 loada 1
27 mod
28 storea 1
29 pop
30 storea 0
31 pop
32 jump loop
33 done: pop
34
35 pop
36
37 // factorial : int -> int
38 // 0! = 1, 4! = 24, ...
39
40
41 // strlen : int* -> int
42 // \0-terminated array
43 loadc 102 // f
44 loadc 111 // o
45 loadc 111 // o
46 loadc 0 // \0
47
48 // allocate : x -> x*int
49
  
```

4

48,20 Ende

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 12
001: LOADC 8
002: LOADA 1
003: JUMPZ 13
004: LOADA 1
005: LOADA 0
006: LOADA 1
007: MOD
008: STOREA 1
009: POP
010: STOREA 0
011: POP
012: JUMP 2
013: POP
014: LOADC 102
015: LOADC 111
016: LOADC 111
017: LOADC 0
  
```

PC: 14
SP: 0
FP: -1
EP: 0
NP: 1000

```

19 // gcd(a, b) = gcd(b, a mod b)
20 loadc 12 // a at 0
21 loadc 8 // b at 1
22 loop: loada 1
23 jumpz done
24 loada 1
25 loada 0
26 loada 1
27 mod
28 storea 1
29 pop
30 storea 0
31 pop
32 jump loop
33 done: pop
34
35 pop
36
37 // factorial : int -> int
38 // 0! = 1, 4! = 24, ...
39
40
41
42 // strlen : int* -> int
43 // \0-terminated array
44 loadc 102 // f
45 loadc 111 // o
  
```

4

19,1 78%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 12
001: LOADC 8
002: LOADA 1
003: JUMPZ 13
004: LOADA 1
005: LOADA 0
006: LOADA 1
007: MOD
008: STOREA 1
009: POP
010: STOREA 0
011: POP
012: JUMP 2
013: POP
014: LOADC 102
015: LOADC 111
016: LOADC 111
017: LOADC 0
  
```

PC: 14
SP: 0
FP: -1
EP: 0
NP: 1000

```

18 // gcd(a, 0) = a
19 // gcd(a, b) = gcd(b, a mod b)
20 //loadc 12 // a at 0
21 //loadc 8 // b at 1
22 //loop: loada 1
23 //jumpz done
24 //loada 1
25 //loada 0
26 //loada 1
27 //mod
28 //storea 1
29 //pop
30 //storea 0
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38
39
40
41 // strlen : int* -> int
42 // \0-terminated array
43 loadc 102 // f
44 loadc 111 // o
  
```

4

44,1 77%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 12
001: LOADC 8
002: LOADA 1
003: JUMPZ 13
004: LOADA 1
005: LOADA 0
006: LOADA 1
007: MOD
008: STOREA 1
009: POP
010: STOREA 0
011: POP
012: JUMP 2
013: POP
014: LOADC 102
015: LOADC 111
016: LOADC 111
017: LOADC 0
  
```

PC: 14
SP: 0
FP: -1
EP: 0
NP: 1000

```

25 //loada 0
26 //loada 1
27 //mod
28 //storea 1
29 //pop
30 //storea 0
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 3 // input: n at 0
39 dup // acc at 1
40
41
42 // strlen : int* -> int
43 // \0-terminated array
44 loadc 102 // f at 0
45 loadc 111 // o
46 loadc 111 // o
47 loadc 0 // \0
48 // strlen 0 = 3
49
50 // allocate : x -> x*int
51
  
```

4

51,0-1 Ende

Select Program

Pfad- oder Ordernamen eingeben:

/home/ttt/Arbeitsfläche/vam/

Filter

Ordner	Dateien
.	ex.cma.swp .vam_prefs README.TXT dependencies.txt ex.cma test.cma vam.bat vam.jar vam.sh

Filternamen eingeben:
ex.cma

OK Filter Abbrechen

Terminal

```

25 //loada 0
26 //loada 1
27 //mod
28 //storea 1
29 //pop
30 //storea 0
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 3 // input: n at 0
39 dup // acc at 1
40 loop: loada 0
41 loadc 1
42 sub
43 dup
44 jumpz done
45 mul
46 jump loop
47
48 done: pop
49 storea 0
50 pop
51
  
```

<ma" 61L, 723C geschrieben 48,9 70%

Select Program

Pfad- oder Ordernamen eingeben:

/home/ttt/Arbeitsfläche/vam/

Filter

Ordner	Dateien
.	ex.cma.swp .vam_prefs README.TXT dependencies.txt ex.cma test.cma vam.bat vam.jar vam.sh

Filternamen eingeben:
ex.cma

OK Filter Abbrechen

Terminal

```

25 //loada 0
26 //loada 1
27 //mod
28 //storea 1
29 //pop
30 //storea 0
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 3 // input: n at 0
39 dup // acc at 1
40 loop: loada 0
41 loadc 1
42 sub
43 storea 0
44 jumpz done
45 loada 0
46 mul
47 jump loop
48
49 done: pop
50 storea 0
51 pop
  
```

<ma" 62L, 736C geschrieben 45,7 68%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 3
001: DUP
002: LOADA 0
003: LOADC 1
004: SUB
005: STOREA 0
006: JUMPZ 10
007: LOADA 0
008: MUL
009: JUMP 2
010: POP
011: STOREA 0
012: POP
013: LOADC 102
014: LOADC 111
015: LOADC 111
016: LOADC 0
  
```

PC: 11
SP: 0
FP: -1
EP: 0
NP: 1000

0

Terminal

```

29 //pop
30 //storea 0
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 3 // input: n at 0
39 dup // acc at 1
40 loada 0
41 jumpz fac0
42 loop: loada 0
43 loadc 1
44 sub
45 storea 0
46 jumpz done
47 loada 0
48 mul
49 jump loop
50 fac0: loadc 1
51 done: storea 0
52 pop
53
54 // strlen : int* -> int
55 // \0-terminated array
  
```

54,13 77%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 3
001: DUP
002: LOADA 0
003: JUMPZ 12
004: LOADA 0
005: LOADC 1
006: SUB
007: STOREA 0
008: JUMPZ 15
009: LOADA 0
010: MUL
011: JUMP 4
012: LOADC 1
013: STOREA 1
014: POP
015: STOREA 0
016: POP
017: LOADC 102
018: LOADC 111
019: LOADC 111
020: LOADC 0
  
```

PC: 1
SP: 0
FP: -1
EP: 0
NP: 1000

3

```

Terminal
Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal
29 //pop
30 //storea 0
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 3 // input: n at 0
39 dup // acc at 1
40 loada 0
41 jumpz fac0
42 loop: loada 0
43 loadc 1
44 sub
45 storea 0
46 jumpz done
47 loada 0
48 mul
49 jump loop
50 fac0: loadc 1
51 storea 1
52 pop
53 done: storea 0
54 pop
55
<ma" 65L, 777C geschrieben 52,3 73%
  
```

Select Program

Pfad- oder Ordernamen eingeben:

/home/ttt/Arbeitsfläche/vam/

Filter

Ordner	Dateien
.	ex.cma.swp .vam_prefs README.TXT dependencies.txt ex.cma test.cma vam.bat vam.jar vam.sh

Filternamen eingeben:
ex.cma

OK Filter Abbrechen

```

Terminal
Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal
29 //pop
30 //storea 0
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 0 // input: n at 0
39 dup // acc at 1
40 loada 0
41 jumpz fac0
42 loop: loada 0
43 loadc 1
44 sub
45 storea 0
46 jumpz done
47 loada 0
48 mul
49 jump loop
50 fac0: loadc 1
51 storea 1
52 pop
53 done: storea 0
54 pop
55
<ma" 65L, 777C geschrieben 38,7 73%
  
```

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 0
001: DUP
002: LOADA 0
003: JUMPZ 12
004: LOADA 0
005: LOADC 1
006: SUB
007: STOREA 0
008: JUMPZ 15
009: LOADA 0
010: MUL
011: JUMP 4
012: LOADC 1
013: STOREA 1
014: POP
015: STOREA 0
016: POP
017: LOADC 102
018: LOADC 111
019: LOADC 111
020: LOADC 0
  
```

PC: 17
SP: 0
FP: -1
EP: 0
NP: 1000

1

```

Terminal
Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 0 // input: n at 0
39 dup // acc at 1
40 loada 0
41 jumpz fac0
42 loop: loada 0
43 loadc 1
44 sub
45 storea 0
46 jumpz done
47 loada 0
48 mul
49 jump loop
50 fac0: loadc 1
51 storea 1
52 pop
53 done: storea 0
54 pop
55
56 // strlen : int* -> int
57 // \0-terminated array
  
```

57,7 78%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 0
001: DUP
002: LOADA 0
003: JUMPZ 12
004: LOADA 0
005: LOADC 1
006: SUB
007: STOREA 0
008: JUMPZ 15
009: LOADA 0
010: MUL
011: JUMP 4
012: LOADC 1
013: STOREA 1
014: POP
015: STOREA 0
016: POP
017: LOADC 102
018: LOADC 111
019: LOADC 111
020: LOADC 0
  
```

PC: 17
SP: 0
FP: -1
EP: 0
NP: 1000

1

```

Terminal
Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
Terminal
31 //pop
32 //jump loop
33 //done: pop
34
35
36 // factorial : int -> int
37 // 0! = 1, 4! = 24, ...
38 loadc 0 // input: n at 0
39 dup // acc at 1
40 loada 0
41 jumpz fac0
42 loop: loada 0
43 loadc 1
44 sub
45 storea 0
46 jumpz done
47 loada 0
48 mul
49 jump loop
50 fac0: loadc 1
51 storea 1
52 pop
53 done: storea 0
54 pop
55
56 // strlen : int* -> int
57 // \0-terminated array
  
```

33,7 78%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Run Nonstop Step back Slow compact heap

```

000: LOADC 0
001: DUP
002: LOADA 0
003: JUMPZ 12
004: LOADA 0
005: LOADC 1
006: SUB
007: STOREA 0
008: JUMPZ 15
009: LOADA 0
010: MUL
011: JUMP 4
012: LOADC 1
013: STOREA 1
014: POP
015: STOREA 0
016: POP
017: LOADC 102
018: LOADC 111
019: LOADC 111
020: LOADC 0
  
```

PC: 17
SP: 0
FP: -1
EP: 0
NP: 1000

1

```

39 //dup // acc at 1
40 //loada 0
41 //jumpz fac0
42 //loop: loada 0
43 //loadc 1
44 //sub
45 //storea 0
46 //jumpz done
47 //loada 0
48 //mul
49 //jump loop
50 //fac0: loadc 1
51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 102 // f at 0
59 loadc 111 // o
60 loadc 111 // o
61 loadc 0 // \0
62 // strlen 0 = 3
63
64 // allocate : x -> x*int
65
  
```

65,0-1 Ende

Select Program

Pfad- oder Ordernamen eingeben:

/home/ttt/Arbeitsfläche/vam/

Filter

Ordner Dateien

.	ex.cma.swp
.	.vam_prefs
.	README.TXT
.	dependencies.txt
.	ex.cma
.	test.cma
.	vam.bat
.	vam.jar
.	vam.sh

Filternamen eingeben:

ex.cma

OK Filter Abbrechen

```

39 //dup // acc at 1
40 //loada 0
41 //jumpz fac0
42 //loop: loada 0
43 //loadc 1
44 //sub
45 //storea 0
46 //jumpz done
47 //loada 0
48 //mul
49 //jump loop
50 //fac0: loadc 1
51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 102 // f at 0
59 loadc 111 // o
60 loadc 111 // o
61 loadc 0 // \0
62 // strlen 0 = 3
63 loadc 0
64
65 // allocate : x -> x*int
  
```

<ma" 66L, 819C geschrieben 63,7 97%

Select Program

Pfad- oder Ordernamen eingeben:

/home/ttt/Arbeitsfläche/vam/

Filter

Ordner Dateien

.	ex.cma.swp
.	.vam_prefs
.	README.TXT
.	dependencies.txt
.	ex.cma
.	test.cma
.	vam.bat
.	vam.jar
.	vam.sh

Filternamen eingeben:

ex.cma

OK Filter Abbrechen

```

40 //loada 0
41 //jumpz fac0
42 //loop: loada 0
43 //loadc 1
44 //sub
45 //storea 0
46 //jumpz done
47 //loada 0
48 //mul
49 //jump loop
50 //fac0: loadc 1
51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 42
59 loadc 102 // f at 0
60 loadc 111 // o
61 loadc 111 // o
62 loadc 0 // \0
63 // strlen 1 = 3
64 loadc 1
65
66 // allocate : x -> x*int
  
```

<ma" 67L, 828C geschrieben 59,19 97%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact heap

```

000: LOADC 42
001: LOADC 102
002: LOADC 111
003: LOADC 111
004: LOADC 0
005: LOADC 1
  
```

PC: 6
SP: 5
FP: -1
EP: 0
NP: 1000

1
0
111
111
102
42

```

41 //jumpz fac0
42 //loop: loada 0
43 //loadc 1
44 //sub
45 //storea 0
46 //jumpz done
47 //loada 0
48 //mul
49 //jump loop
50 //fac0: loadc 1
51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 42
59 loadc 102 // f at 1
60 loadc 111 // o
61 loadc 111 // o
62 loadc 0 // \0
63 // strlen 1 = 3
64 loadc 1 // starting address
65
66 // allocate : x -> x*int
67
  
```

67,0-1 Ende

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

000: LOADC 42
001: LOADC 102
002: LOADC 111
003: LOADC 111
004: LOADC 0
005: LOADC 1

```

PC: 6
SP: 5
FP: -1
EP: 0
NP: 1000

```

44 //sub
45 //storea 0
46 //jumpz done
47 //loada 0
48 //mul
49 //jump loop
50 //fac0: loadc 1
51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 42
59 loadc 102 // f at 1
60 loadc 111 // o
61 loadc 111 // o
62 loadc 0 // \0
63 // strlen 1 = 3
64 loadc 1 // starting address
65
66 // allocate : x -> x*int
67 loadc 3
68 loop: loada 0
69 jumpz done
70

```

1
0
111
111
102
42

-- EINFÜGEN -- 70,1 Ende

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

000: LOADC 42
001: LOADC 102
002: LOADC 111
003: LOADC 111
004: LOADC 0
005: LOADC 1

```

PC: 6
SP: 5
FP: -1
EP: 0
NP: 1000

```

48 //mul
49 //jump loop
50 //fac0: loadc 1
51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 42
59 loadc 102 // f at 1
60 loadc 111 // o
61 loadc 111 // o
62 loadc 0 // \0
63 // strlen 1 = 3
64 loadc 1 // starting address
65
66 // allocate : x -> x*int
67 loadc 3
68 loop: loada 0
69 jumpz done
70 loada 0
71 loadc 1
72 sub
73
74 done:

```

1
0
111
111
102
42

-- EINFÜGEN -- 74,1 Ende

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

000: LOADC 42
001: LOADC 102
002: LOADC 111
003: LOADC 111
004: LOADC 0
005: LOADC 1

```

PC: 6
SP: 5
FP: -1
EP: 0
NP: 1000

```

51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 42
59 loadc 102 // f at 1
60 loadc 111 // o
61 loadc 111 // o
62 loadc 0 // \0
63 // strlen 1 = 3
64 loadc 1 // starting address
65
66 // allocate : x -> x*int
67 loadc 3
68 loop: loada 0
69 jumpz done
70 loada 0
71 loadc 1
72 sub
73 storea 0
74 pop
75 loadc 0
76 jump loop
77 done:

```

1
0
111
111
102
42

-- EINFÜGEN -- 77,6 Ende

Select Program

Pfad- oder Ordernamen eingeben:

/home/ttt/Arbeitsfläche/Vam/

act heap

Filter

Ordner

Dateien

- ex.cma.swp
- .vam_prefs
- README.TXT
- dependencies.txt
- ex.cma
- test.cma
- vam.bat
- vam.jar
- vam.sh

Filternamen eingeben:

ex.cma

OK Filter Abbrechen

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Restart Nonstop Step back Slow compact hea

```

51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 42
59 loadc 102 // f at 1
60 loadc 111 // o
61 loadc 111 // o
62 loadc 0 // \0
63 // strlen 1 = 3
64 loadc 1 // starting address
65 dup // current address
66 loop: dup
67 load
68 jumpz done
69 loadc 1
70 add
71 jump loop
72 done: sub
73 neg
74
75
76 // allocate : x -> x*int
77 //loadc 3

```

<a" 87L, 1045C geschrieben 66,6 83%

VAM - Version 2.0 Abstrakt M

VAM Run Options ?

Program Pause Nonstop Step back Slow compact heap

```
000: LOADC 42
001: LOADC 102
002: LOADC 111
003: LOADC 111
004: LOADC 0
005: LOADC 1
006: DUP
007: DUP
008: LOAD
009: JUMPZ 13
010: LOADC 1
011: ADD
012: JUMP 7
013: SUB
014: NEG
```

PC: 14
SP: 6
FP: -1
EP: 0
NP: 1000

1	4
0	
111	
111	
102	
42	

Terminal

```
51 //storea 1
52 //pop
53 //done: storea 0
54 //pop
55
56 // strlen : int* -> int
57 // \0-terminated array
58 loadc 42
59 loadc 102 // f at 1
60 loadc 111 // o
61 loadc 111 // o
62 loadc 0 // \0
63 // strlen 1 = 3
64 loadc 1 // starting address
65 dup // current address
66 loop: dup
67 load
68 jumpz done
69 loadc 1
70 add
71 jump loop
72 done: sub
73 neg
74
75
76 // allocate : x -> x*int
77 //loadc 3
<a" 87L, 1045C geschrieben 66,6    83%
```