| **VMコマンド** | **疑似コード** | **アセンブリ** |
| --- | --- | --- |
| call f n | push return-address | // D←return-address  @return-address[[1]](#footnote-1)  D=A  // M[SP]←D  @SP  A=M  M=D  // SP++  @SP  M=M+1 |
| push LCL | // D←@LCL  @LCL  D=M  // M[SP]←D  @SP  A=M  M=D  // SP++  @SP  M=M+1 |
| push ARG | // D←@ARG  @LCL  D=M  // M[SP]←D  @SP  A=M  M=D  // SP++  @SP  M=M+1 |
| push THIS | // D←@THIS  @LCL  D=M  // M[SP]←D  @SP  A=M  M=D  // SP++  @SP  M=M+1 |
| push THAT | // D←@THAT  @LCL  D=M  // M[SP]←D  @SP  A=M  M=D  // SP++  @SP  M=M+1 |
| ARG = SP – n – 5 | // D←SP – n - 5  @SP  D=M  @{n+5}[[2]](#footnote-2)  D=D-A  // ARG←D  @ARG  M=D |
| LCL = SP | @SP  D=M  @LCL  M=D |
| goto f | @f[[3]](#footnote-3)  0;JMP |
| (return-address) | (return-address)[[4]](#footnote-4) |
| function f k | (f) | (f) |
| repeat k[[5]](#footnote-5) time: | for loop 5  // D←0  @0  D=A  // M[SP]←D  @SP  A=M  M=D  // SP++  @SP  M=M+1 |
| push 0 |
| return | FRAME = LCL | // R13をFRAMEとする  @LCL  D=M  @R13  M=D |
| RET = \*(FRAME – 5) | // R14をRETとする  @5  A=D-A  D=M  @R14  M=D |
| \*ARG = POP() | // SP=SP-1  @SP  M=M-1  // D←M[SP]  A=M  D=M  // M[ARG]←D  @ARG  A=M  M=D |
| SP = ARG + 1 | @ARG  D=M+1  @SP  M=D |
| THAT = \*(FRAME – 1) | @R13  D=M  @1  A=D-A  D=M  @THAT  M=D |
| THIS = \*(FRAME – 2) | @R13  D=M  @2  A=D-A  D=M  @THIS  M=D |
| ARG = \*(FRAME – 3) | @R13  D=M  @3  A=D-A  D=M  @ARG  M=D |
| LCL = \*(FRAME – 4) | @R13  D=M  @4  A=D-A  D=M  @LCL  M=D |
| goto RET | @R14  A=M  0;JMP |

# SimpleFunctionテストメモ

## 初期値

|  |  |  |
| --- | --- | --- |
| RAM[0] | SP | 317 |
| RAM[1] | LCL | 317 |
| RAM[2] | ARG | 310 |
| RAM[3] | THIS | 3000 |
| RAM[4] | THAT | 4000 |
| RAM[310] | ARG[0] | 1234 |
| RAM[311] | ARG[1] | 37 |
| RAM[312] | リターンアドレス | 1000 |
| RAM[313] | 保存されたLCL | 305 |
| RAM[314] | 保存されたARG | 300 |
| RAM[315] | 保存されたTHIS | 3010 |
| RAM[316] | 保存されたTHAT | 4010 |

## 予想結果

|  |  |  |
| --- | --- | --- |
| RAM[0] | SP | 311 |
| RAM[1] | LCL | 305 |
| RAM[2] | ARG | 300 |
| RAM[3] | THIS | 3010 |
| RAM[4] | THAT | 4010 |
| RAM[310] | ARG[0] | 1196 |

# ブートストラップコード

|  |  |
| --- | --- |
| SP=256 | @256  D=A  @SP  M=D |
| call Sys.init | @Sys.init  0;JMP |

1. return-address文字列を生成する必要あり。$ret.x [↑](#footnote-ref-1)
2. callコマンドのnを覚えておく必要あり [↑](#footnote-ref-2)
3. callコマンドのfを覚えておく必要あり [↑](#footnote-ref-3)
4. 先に作ったラベル名をここでも使う [↑](#footnote-ref-4)
5. functionコマンドのkを覚えておく必要あり [↑](#footnote-ref-5)