add:
```
@SP
A = M-1
.D = M
A = A-1
M = D+M
@SP
M = M-1
```

Sub:
```
@SP
A = M-1
D = M
A = A-1
M = M-D
@SP
M = M-1
```

Deg:
```
@SP
A = M-1
M = -M
```

eq:
```
@SP
A = M-1
D = M
A = A-1
D = D-M
@EQ;
D; JEQ
@SP
A = M-1
A = A-1
M = 0
@END-E;
0; JMP
(EQ;)
@SP
A = M-1
A = A-1
M = 1
(END-E;)
@SP
M = M-1
```

gt:
```
@SP
A = M-1
D = M
A = A-1
D = M-D
@GT;
D; JGT
@SP
A = M-1
A = A-1
M = 0
@END-G;
0; JMP
(GT;)
@SP
A = M-1
A = A-1
M = 1
(END-G;)
@SP
M = M-1
```

Dot:
```
@SP
A = M-1
D = M
@TRUE-N;

D; JGT
D; JLT
@SP
A = M-1
@END-N;
0; JMP
(TRUE-N;)
@SP
A = M-1
M = 0

(END-N;)
```

lt:
```
@SP
A = M-1
D = M
A = A-1
D = D-M
@LT;
D; JGT
@SP
A = M-1
A = A-1
M = 0
@END-L;
0; JMP
(LT;)
@SP
A = M-1
A = A-1
M = 1
(END-L;)
@SP
M = M-1
```

and:
```
@SP
A = M-1
D = M
@FALSE-A;
D; JEQ
@SP
A = M-1
A = A-1
D = M
@FALSE-A;
D; JEQ
@SP
A = M-1
A = A-1
M = 1
@END-A;
0; JMP
(FALSE-A;)
@SP
A = M-1
A = A-1
M = 0
(END-A;)
@SP
M = M-1
```

or:
```
@SP
A = M-1
D = M
@TRUE-O;
D; JGT
D; JLT
@SP
A = M-1
A = A-1
D = M
@TRUE-O;
D; JGT
D; JLT
@SP
A = M-1
A = A-1
M = 0
@END-O;
0; JMP
(TRUE-O;)
@SP
A = M-1
A = A-1
M = 1
(END-O;)
@SP
M = M-1
```

תרגום פקודות pop/push ∫ HACK:

pop local X:
```
@X
D=A
@LCL
D=D+M
@13
M=D
@SP
A=M-1
D=M
@13
A=M
M=D
@SP
M=M-1
```

⊛ כל ה"וכו' כנ"ל
pop argument X
pop this X
pop that X

רק שמקום LCL יהיה:
ARG
THIS
THAT

push local X:
```
@X
D=A
@LCL
A=D+M
D=M
@SP
A=M
M=D
@SP
M=M+1
```

⊛ כל ה"וכו' כנ"ל
push argumln X:
push this X:
push that X:

רק שמקום Lcl יהיה:
ARG
THIS
THAT

push constant X:
```
@X
D=A
@SP
A=M
M=D
@SP
M=M+1
```

pop temp X:
```
@X
D=A
@5
D=D+A
@13
M=D
@SP
A=M-1
D=M
@    13
A=M
M=D
@SP
M=M-1
```

push temp X:
```
@X
D=A
@5
A=D+A
D=M
@SP
A=M
M=D
@SP
M=M+1
```

push static X:
```
@f.X
D=M
@SP
A=M
M=D
@SP
M=M+1
```

pop Static X:
```
@SP
A=M-1
D=M
@f.X
M=D
@SP
M=M-1
```

pop pointer 0:
```
@SP
A=M-1
D=M
@3
M=D
@SP
M=M-1
```

push pointer 0:
```
@3
D=M
@SP
A=M
M=D
@SP
M=M+1
```

pop pointer 1:
```
@SP
A=M-1
D=M
@4
M=D
@SP
M=M-1
```

push pointer 1:
```
@4
D=M
@SP
A=M
M=D
@SP
M=M+1
```