STACK ABSTRACT MACHINE (SAM)

How SAM works (high level)

command-file

```
PUSHIMM 2
PUSHIMM 3
ADD
:
```

load file into SAM

loader (command-file)

```
Loc = 0;
Open command-file for input;
while (!EOF) {
    Read a command from file
    Store command in Program[Loc];
    Loc ++;
}
```

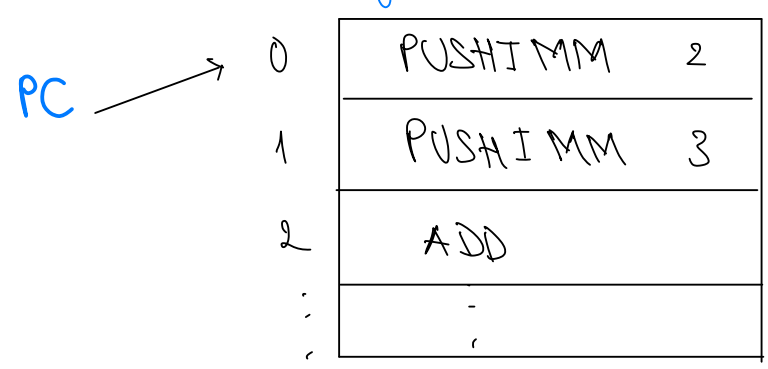
Interpreter

```
PC = 0;
while (! HALT)
    execute command Program[PC]
```

HALT

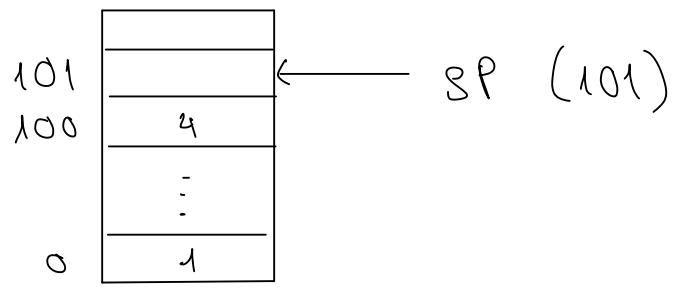
SAM

Program area

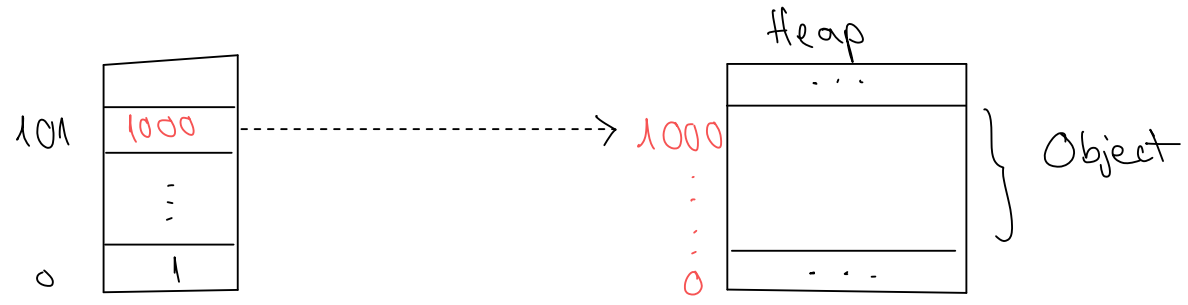


SAM is a Stack Machine

- All data stored in the stack or heap
- Stack pointer (SP) points to the first free location in the stack

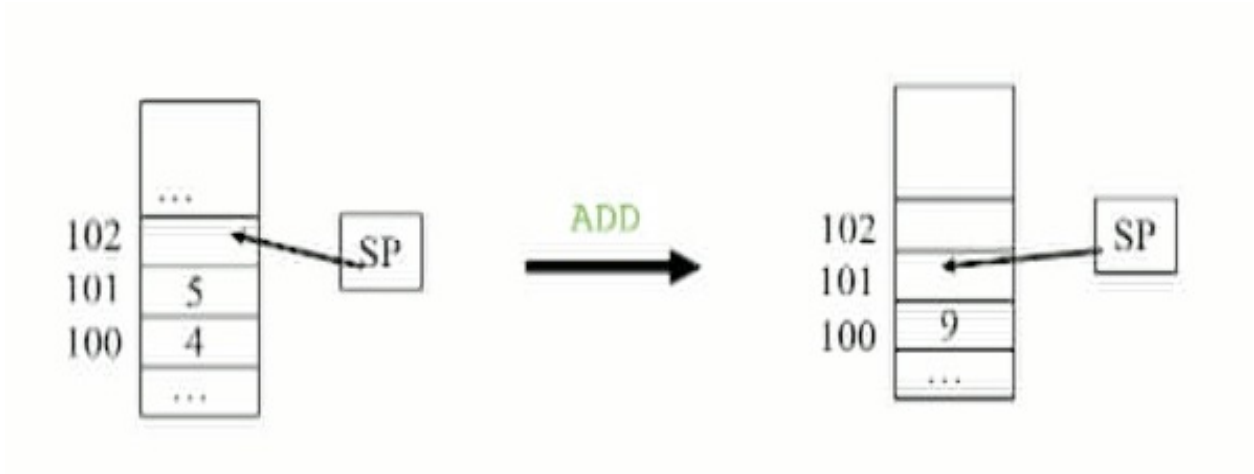


- Atomic types (int, bool) and addresses take 1 stack location
- Strings and Objects are stored in the heap



Some common operations

ADD



PUSHIMM [c]

ALU Commands:

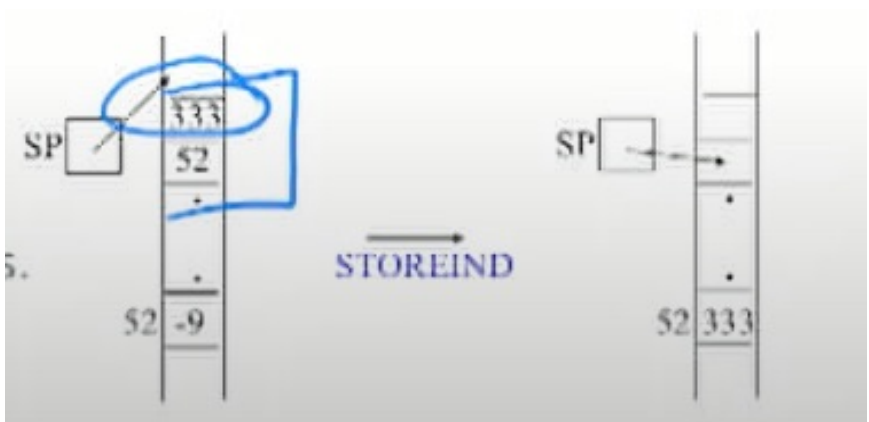
- `ADD`, `SUB`, ...
- `DUP`: duplicate TOS.
- `ISPOS` ("is positive"):
  - Pop stack; let popped value be *Vt*.
  - If *Vt* is positive, push `true` (1); otherwise push `false` (0).
- `ISNEG` ("is negative"): Same, but tests for negative value on TOS.
- `ISNIL` ("is null"): Same, but tests for zero value on TOS.
- `CMP`:
  - Pop two values *Vt* and *Vb* from stack.
  - If (*Vb* < *Vt*), push 1.
  - If (*Vb* = *Vt*), push 0.
  - If (*Vb* > *Vt*), push -1.

Load / Store Commands:

- Indirect mode:

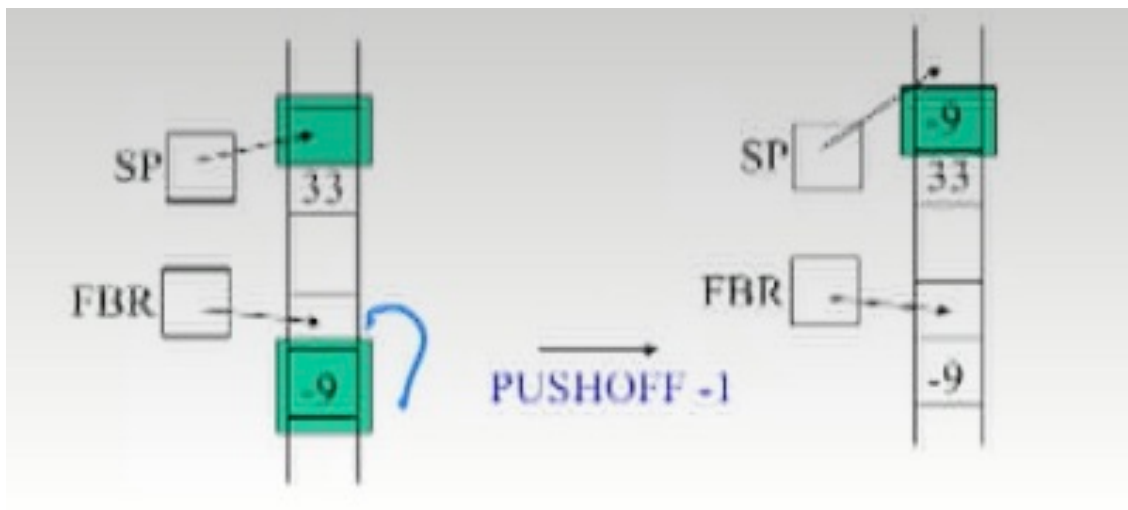
PUSH IND

STOREIND



- Offset mode:

PUSHOFF [n]



STORE [n]