

Instructions

- load (ld)

Can only be used to access memory.

ld 4, R4→ copies value from memory address 4 to R4

ld R3, R4→ copies value from memory address value of R3 to R4

- move (mov)

Can only be used between registers or to store values in registers.

mov 4, R4→ moves the value 4 to R4. mov R2, R4→ copies the value in R2 to R4.

- store (str)

str 4, R4→ stores value from R4 to memory address 4.

str R3, R4→ stores value from R4 to memory address at R3.

- add (add)

Has 3 registers. First one and second one are the operands, the last one is where the result will be saved.

add R4, R3, R4→ takes values from R4 and R3, sums them and saves the result in R4.

- subtract (sub)

Uses 3 registers. First one and second one are the operands, the last one is where the result will be saved.

sub R4, R3, R4→ takes values from R4 and R3, subtracts them and saves the result in R4.

- Multiply (mult)

Uses 3 registers. First one and second one are the operands, the last one is where the result will be saved.

mult R4, R3, R4→ takes values from R4 and R3, multiplies them and saves the result in R4.

- compare (cmp) Takes two registers and compares the values in them, sets a flag to the value after subtracting the two.

cmp R2, R1→ takes values from R2 and R1, computes the subtraction (value in R1 - value in R2) and the result is saved on the flag.

- jump (jmp)

Instruction that branches to a section.

```

1 loop:
2     ...some code...
3     jmp loop

```

- jump equal (je)

Instruction that branches to a section only if the compare flag is set to 0.

```

1 loop:
2     ...some code...
3     cmp R2,R3
4     je loop

```

- jump above (ja)

Instruction that branches to a section only if the compare flag is set to a number greater than 0.

```

1 loop:
2     ...some code...
3     cmp R2,R3
4     ja loop

```

- jump below (jb)

Instruction that branches to a section only if the compare flag is set to a number smaller than 0.

```

1 loop:
2     ...some code...
3     cmp R2,R3
4     jb loop

```

- end (end)

Instruction set at the end of the code to let the simulator know all instructions have ended.