

```
// ***** INSTRUCTIONS *****

void load() {
    printf("\tLoad\n");
    if (mode == 1) {
        Registers[reg] = address;
    }
    else {
        Registers[reg] = (short)mainMemory[address];
    }

    changeCondition(reg);
}

void store() {
    printf("\tStore\n");
    mainMemory[address] = (unsigned short)Registers[reg];

    changeCondition(reg);
}

void add() {
    printf("\tAdd\n");
    if (mode == 1) {
        Registers[reg] = Registers[0] + address;
    }
    else {
        Registers[reg] = Registers[0] + (short)mainMemory[address];
    }

    changeCondition(reg);
}

void sub() {
    printf("\tSubtract\n");
    if (mode == 1) {
        Registers[reg] = Registers[0] - address;
    }
    else {
        Registers[reg] = Registers[0] - (short)mainMemory[address];
    }

    changeCondition(reg);
}

void adr() {
    printf("\tAdd Register\n");
    Registers[0] = Registers[0] + Registers[reg];

    changeCondition(0);
}
```

```
void sur() {
    printf("\tSubtract Register\n");
    Registers[0] = Registers[0] - Registers[reg];

    changeCondition(0);
}

void and() {
    printf("\tAnd\n");
    if (mode == 1) {
        Registers[reg] = Registers[0] & address;
    }
    else {
        Registers[reg] = Registers[0] & (short)mainMemory[address];
    }

    changeCondition(reg);
}

void or() {
    printf("\tOr\n");
    if (mode == 1) {
        Registers[reg] = Registers[0] | address;
    }
    else {
        Registers[reg] = Registers[0] | (short)mainMemory[address];
    }

    changeCondition(reg);
}

void not() {
    printf("\tNot\n");

    Registers[reg] = ~Registers[reg];

    changeCondition(reg);
}

void jmp() {
    printf("\tJump\n");
    PC = (unsigned short)address;
}

void jeq() {
    printf("\tJump Equal\n");
    if (CC == 2) PC = (unsigned short)address;
}

void jgt() {
    printf("\tJump Greater\n");
    if (CC == 1) PC = (unsigned short)address;
```

```
}

void jlt() {
    printf("\tJump Less\n");
    if (CC == 4) PC = (unsigned short)address;
}

void compare() {
    printf("\tCompare\n");
    if (Registers[reg] > 0) {
        CC = 1;
    }
    else if(Registers[reg] == 0) {
        CC = 2;
    }
    else if(Registers[reg] < 0) {
        CC = 4;
    }
    else {}
}

void clear() {
    printf("\tClear\n");
    Registers[reg] = 0;

    changeCondition(reg);
}

void halt() {
    haltFlag = true;
    printf("\tHalt\n");

    printf("Execution complete.\n");
}
```