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

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

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

   changeCondition(reg);
   programClock++;
}

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

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

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

   changeCondition(0);
   programClock++;
}
```

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

   changeCondition(0);
   programClock++;
}

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

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

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

   Registers[reg] = ~Registers[reg];

   changeCondition(reg);
   programClock++;
}

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

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

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

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

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

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

   changeCondition(reg);
   programClock++;
}

void halt() {
   haltFlag = true;
   printf("\tHalt\n");
   printf("Execution complete.");
   programClock++;
}
```