

First we create a file that takes the path of the file we will read. Then we go through the contents of this file with the Scanner.

```
Scanner siteler = new Scanner(file);
while (siteler.hasNextLine()) {
    String str = siteler.nextLine().toUpperCase().trim();
```

We remove the empty lines and extra spaces and add the line to the ArrayList we created.

```
while (siteler.hasNextLine()) {
    String str = siteler.nextLine().toUpperCase().trim();
    if (!str.contains("%")) {
        while (str.contains(" ")) str = str.replace(target: " ", replacement: " ");
        str = str.substring(str.indexOf(" ") + 1);
        if (!str.equals("")) commands.add(str);
    }
}
```

Then we create the array and variables we need

```
boolean isStart = false;
int AC = 0;
int[] M = new int[256];
int E = 0;
```

We then create a 'for' loop. In this loop, we go through each line in our list one by one. Here we keep the first part of each command as 'baseCommand' and the other part as 'commandNum'.

Example:

Command: LOAD 20

baseCommand: LOAD

commandNum: 20

```
for (int PC = 0; PC < commands.size(); PC++) {
    String baseCommand = commands.get(PC);
    int commandNum = 0;
    if (commands.get(PC).contains(" ")) {
        baseCommand = commands.get(PC).substring(0, commands.get(PC).indexOf(" "));
        commandNum = Integer.parseInt(commands.get(PC).substring(commands.get(PC).indexOf(" ") + 1));
    }
}
```

Then, with the 'if statement', we check whether the program has started or not. If it has started before, it will continue. If not, the current command should be 'START' to continue.

```
for (int PC = 0; PC < commands.size(); PC++) {
    String baseCommand = commands.get(PC);
    int commandNum = 0;
    if (commands.get(PC).contains(" ")) {
        baseCommand = commands.get(PC).substring(0, commands.get(PC).indexOf(" "));
        commandNum = Integer.parseInt(commands.get(PC).substring(commands.get(PC).indexOf(" ") + 1));
    }
    if (isStart || baseCommand.equals("START"))
        switch (baseCommand) {
            case "START":
                isStart = true;
                break;
        }
}
```

Then we define command operations with 'switch case'

```
if (isStart || baseCommand.equals("START"))
    switch (baseCommand) {
        case "START":
            isStart = true;
            break;
        case "LOAD":
            AC = commandNum;
            break;
        case "LOADM":
            AC = M[commandNum];
            break;
        case "STORE":
            M[commandNum] = AC;
            break;
        case "CHPM":
            if (AC > M[commandNum]) F = 1;
            else if (AC < M[commandNum]) F = -1;
            else F = 0;
            break;
        case "CJMP":
            if (F > 0) PC = commandNum - 1;
            break;
        case "JMP":
            PC = commandNum - 1;
            break;
        case "ADD":
            AC = AC + commandNum;
            break;
        case "ADDM":
            AC = AC + M[commandNum];
            break;
        case "SUBM":
            AC -= M[commandNum];
            break;
        case "SUB":
            AC -= commandNum;
            break;
        case "MUL":
            AC *= commandNum;
            break;
        case "MULM":
            AC *= M[commandNum];
            break;
        case "DISP":
            System.out.println(AC);
            break;
        case "HALT":
            isStart = false;
            break;
    }
```