```
/* COP3402 Systems Software PM/0 Virtual Machine
   Coded by Harshika Jindal
   University of Central Florida
   PID: 5521417
*/

#include <stdio.h>
#include <stdlib.h>

#define ARRAY_SIZE 500

// Instruction Register structure
typedef struct {
    int op;
    int l;
    int m;
} Instruction;

// Global Variables
int pas[ARRAY_SIZE] = {0};
int pc = 10;
int bp = 499;
int sp = 500;
Instruction ir;
int halt = 1;

// Recursive function to print activation records
// void printStack(int spIndex, int currentBP) {
//     if (spIndex >= ARRAY_SIZE) return;
//     if (spIndex == currentBP && currentBP != 499) {
//         ;
//     }
//     if (spIndex <= ARRAY_SIZE) {
//         printf("%d\t%d", pas[currentBP],pas[spIndex]);
//         printStack(spIndex + 1, currentBP);
//     }
// }
// Function to print current machine state
void printState() {
    printf(" %d\t%d\t%d\t%d\t%d\t",ir.l, ir.m, pc,bp,sp);

    for (int i = 499; i >= sp-1; i--) {
        if(i == bp && bp != 499) printf("| ");
        if(i >= ARRAY_SIZE - (ARRAY_SIZE- sp)) printf("%d ", pas[i]);
    }

    printf("\n");
}

// Find base L levels down
int base(int bp, int l) {
    int arb = bp;
    while (l > 0) {
        arb = pas[arb];
        l--;
    }
    return arb;
}
```

```
int main(int argc, char *argv[]) {
    if (argc < 2) {
        printf("Usage: %s <input_file>\n", argv[0]);
        return 1;
    }

    FILE *fp = fopen(argv[1], "r");
    if (!fp) {
        perror("Error opening input file");
        return 1;
    }

    int index = 10;
    while (fscanf(fp, "%d %d %d", &pas[index], &pas[index + 1], &pas[index + 2]) ==
3) {
        index += 3;
    }
    fclose(fp);

    printf("\n                PC\tBP\tSP\tstack\n");

    if (sp >= ARRAY_SIZE) {
        printf("Initial Values: %d\t%d\t%d\n", pc, bp, sp);
    }

    while (halt) {

        // FETCH instruction
        ir.op = pas[pc];
        ir.l = pas[pc + 1];
        ir.m = pas[pc + 2];

        //printState();
        pc += 3;

        // EXECUTE instruction
        switch (ir.op) {
            case 1:
                printf("INC");
                sp -= ir.m;
                printState();
                break;

            case 2:
                printf("RTN");
                switch (ir.m) {
                    case 0:
                        sp = bp + 1;
                        bp = pas[sp - 2];
                        pc = pas[sp - 3];
                        break;
                    case 1:
                        pas[sp + 1] = pas[sp + 1] + pas[sp];
                        sp++;
                        break;
                    case 2:
                        pas[sp + 1] = pas[sp + 1] - pas[sp];
                        sp++;
                        break;
```

```
            case 3:
                pas[sp + 1] = pas[sp + 1] * pas[sp];
                sp++;
                break;
            case 4:
                pas[sp + 1] = pas[sp + 1] / pas[sp];
                sp++;
                break;
            case 5:
                pas[sp + 1] = pas[sp + 1] == pas[sp];
                sp++;
                break;
            case 6:
                pas[sp + 1] = pas[sp + 1] != pas[sp];
                sp++;
                break;
            case 7:
                pas[sp + 1] = pas[sp + 1] < pas[sp];
                sp++;
                break;
            case 8:
                pas[sp + 1] = pas[sp + 1] <= pas[sp];
                sp++;
                break;
            case 9:
                pas[sp + 1] = pas[sp + 1] > pas[sp];
                sp++;
                break;
            case 10:
                pas[sp + 1] = pas[sp + 1] >= pas[sp];
                sp++;
                break;
        }
        printState();
        break;
    case 3:
    printf("LOD");
        sp--;
        pas[sp] = pas[base(bp, ir.l) - ir.m];
        printState();
        break;
    case 4:
    printf("STO");
        pas[base(bp, ir.l) - ir.m] = pas[sp];
        sp++;
        printState();
        break;
    case 5:
    printf("CAL");
        pas[sp - 1] = base(bp, ir.l);
        pas[sp - 2] = bp;
        pas[sp - 3] = pc;
        bp = sp - 1;
        pc = ir.m;
        printState();
        break;
    case 6:
    printf("LIT");
        sp--;
```

```
                pas[sp] = ir.m;
                printState();
                break;
            case 7:
            printf("JMP");
                pc = ir.m;
                printState();
                break;
            case 8:
            printf("JPC");
                if (pas[sp] == 0)
                    pc = ir.m;
                sp++;
                printState();
                break;
            case 9:

                if (ir.m == 1) {
                    printf("Output result is: %d\n", pas[sp]);
                    sp++;
                } else if (ir.m == 2) {
                    int input;
                    printf("Please Enter an Integer: ");
                    scanf("%d", &input);
                    sp--;
                    pas[sp] = input;
                } else if (ir.m == 3) {
                    halt = 0;
                }
                printf("SYS");
                printState();
                break;
        }
    }

    return 0;
}
```