```c
#include"simulator.h"



void run(int * program_begin, int * program_end, void * memory, cpu_t * cpu)
{
    printf("%X %X\n", *program_begin, *program_end);
    cpu->PC = *program_begin;
    int temp = 0;
    int temp2 = 0;
    int temp3 = 0;
    int x_mod = 0;
    FILE * input = NULL;
    FILE * output = NULL;
    input = fopen("F1.txt", "r");
    output = fopen("F5.txt", "w");
    cpu->L = *program_end;


    while(cpu->PC < *program_end)
    {
        x_mod = 0;
        temp = 0;
        temp2 = 0;
        temp3 = 0;
        if(*((uint8_t *)memory + cpu->PC + 1) >= 128)
        {
            x_mod = 1;
        }

        switch(*((uint8_t *)memory + cpu->PC))
        {
            case 24:
                printf("ADD     ");
                temp = *((uint8_t *)memory + cpu->PC + 1);
                temp = temp << 8;
                temp |= *((uint8_t *)memory + cpu->PC + 2);

                temp2 = *((uint8_t *)memory + temp + 2);
                cpu->A += temp2;

                temp2 = *((uint8_t *)memory + temp + 1);
                temp2 = temp2 << 8;
                cpu->A += temp2;

                temp2 = *((uint8_t *)memory + temp);
                temp2 = temp2 << 16;
                cpu->A += temp2;


                cpu->PC += 3;
                break;

            case 52:
                printf("JGT     ");
                cpu->PC += 3;
                break;

            case 0:
                printf("LDA     ");
                temp = *((uint8_t *)memory + cpu->PC + 1);
                temp = temp << 8;
                temp |= *((uint8_t *)memory + cpu->PC + 2);

                if(*((uint8_t *)memory + temp) == '.')
                {
                    cpu->A = 0;
                }
                else
                {
                    cpu->A = *((uint8_t *)memory + temp);
                }

                cpu->A = cpu->A << 8;

                if(*((uint8_t *)memory + temp + 1) == '.')
                {
                    cpu->A |= 0;
```

```
            }
            else
            {
                cpu->A |= *((uint8_t *)memory + temp + 1);
            }

            cpu->A = cpu->A << 8;

            if(*((uint8_t *)memory + temp + 2) == '.')
            {
                cpu->A |= 0;
            }
            else
            {
                cpu->A |= *((uint8_t *)memory + temp + 2);
            }



            cpu->PC += 3;
            break;

        case 4:
            printf("LDX     ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            if(*((uint8_t *)memory + temp) == '.')
            {
                cpu->X = 0;
            }
            else
            {
                cpu->X = *((uint8_t *)memory + temp);
            }

            cpu->X = cpu->X << 8;

            if(*((uint8_t *)memory + temp + 1) == '.')
            {
                cpu->X |= 0;
            }
            else
            {
                cpu->X |= *((uint8_t *)memory + temp + 1);
            }

            cpu->X = cpu->X << 8;

            if(*((uint8_t *)memory + temp + 2) == '.')
            {
                cpu->X |= 0;
            }
            else
            {
                cpu->X |= *((uint8_t *)memory + temp + 2);
            }


            cpu->PC += 3;
            break;

        case 216:
            printf("RD      ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);


            //cpu->A = *((uint8_t *)memory + temp);

            temp2 = getc(input);
            if(temp2 == -1)
            {
                cpu->A = 0;
            }
            else
            {
```

```
                cpu->A = (unsigned int)temp2;
            }



            cpu->PC += 3;
            break;

        case 220:
            printf("WD      ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            //*((uint8_t *)memory + temp) = cpu->A;
            putc(cpu->A, output);

            cpu->PC += 3;
            break;

        case 84:
            printf("STCH    ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            if(x_mod == 1)
            {
                temp = temp ^ (1 << 15);
                *((uint8_t *)memory + temp + cpu->X) = cpu->A;
            }
            else
            {
                *((uint8_t *)memory + temp) = cpu->A;
            }

            cpu->PC += 3;
            break;

        case 16:
            printf("STX     ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp  << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            temp2 = cpu->X;

            *((uint8_t *)memory + temp + 2) = temp2;
            temp2 = temp2 >> 8;
            *((uint8_t *)memory + temp + 1) = temp2;
            temp2 = temp2 >> 8;
            *((uint8_t *)memory + temp) = temp2;


            cpu->PC += 3;

            break;

        case 12:
            printf("STA     ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp  << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            temp2 = cpu->A;

            *((uint8_t *)memory + temp + 2) = temp2;
            temp2 = temp2 >> 8;
            *((uint8_t *)memory + temp + 1) = temp2;
            temp2 = temp2 >> 8;
            *((uint8_t *)memory + temp) = temp2;


            cpu->PC += 3;

            break;

        case 44:
            printf("TIX     ");
```

```
                cpu->X++;

                temp = *((uint8_t *)memory + cpu->PC + 1);
                temp = temp << 8;
                temp |= *((uint8_t *)memory + cpu->PC + 2);

                temp2 = *((uint8_t *)memory + temp);
                temp2 = temp2 << 8;
                temp2 |= *((uint8_t *)memory + temp + 1);
                temp2 = temp2 << 8;
                temp2 |= *((uint8_t *)memory + temp + 2);


                if(cpu->X == temp2)
                {
                    cpu->SW = '=';
                }
                else if(cpu->X < temp2)
                {
                    cpu->SW = '<';
                }
                else
                {
                    cpu->SW = '>';
                }

                cpu->PC += 3;
                break;

            case 64:
                printf("AND     ");
                cpu->PC += 3;
                break;

            case 60:
                printf("J       ");
                temp = cpu->PC;
                cpu->PC = *((uint8_t *)memory + temp + 1);
                cpu->PC = cpu->PC  << 8;
                cpu->PC |= *((uint8_t *)memory + temp + 2);
                break;

            case 56:
                printf("JLT     ");
                temp = cpu->PC;
                if(cpu->SW == '<')
                {
                    cpu->PC = *((uint8_t *)memory + temp + 1);
                    cpu->PC = cpu->PC  << 8;
                    cpu->PC |= *((uint8_t *)memory + temp + 2);

                }
                else
                {
                    cpu->PC += 3;
                }
                break;

            case 80:
                printf("LDCH    ");
                temp = *((uint8_t *)memory + cpu->PC + 1);
                temp = temp << 8;
                temp |= *((uint8_t *)memory + cpu->PC + 2);

                if(x_mod == 1)
                {
                    temp = temp ^ (1 << 15);
                    cpu->A = *((uint8_t *)memory + temp + cpu->X);
                }
                else
                {
                    cpu->A = *((uint8_t *)memory + temp);
                }

                cpu->PC += 3;
                break;

            case 32:
                printf("MUL     ");
```

```
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            temp3 = cpu->A;
            cpu->A = 0;

            temp2 = *((uint8_t *)memory + temp + 2);
            cpu->A += temp3 * temp2;

            temp2 = *((uint8_t *)memory + temp + 1);
            temp2 = temp2 << 8;
            cpu->A += temp3 * temp2;

            temp2 = *((uint8_t *)memory + temp);
            temp2 = temp2 << 16;
            cpu->A += temp3 * temp2;


            cpu->PC += 3;
            break;

        case 76:
            printf("RSUB    ");
            cpu->PC = cpu->L;
            break;

        case 20:
            printf("STL     ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp  << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            temp2 = cpu->L;

            *((uint8_t *)memory + temp + 2) = temp2;
            temp2 = temp2 >> 8;
            *((uint8_t *)memory + temp + 1) = temp2;
            temp2 = temp2 >> 8;
            *((uint8_t *)memory + temp) = temp2;



            cpu->PC += 3;
            break;

        case 28:
            printf("SUB     ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            temp2 = *((uint8_t *)memory + temp);
            temp2 = temp2 << 8;

            temp2 = *((uint8_t *)memory + temp + 1);
            temp2 = temp2 << 8;

            temp2 = *((uint8_t *)memory + temp + 2);

            cpu->A = cpu->A - temp2;


            cpu->PC += 3;
            break;



        case 40:
            printf("COMP    ");
            temp = *((uint8_t *)memory + cpu->PC + 1);
            temp = temp << 8;
            temp |= *((uint8_t *)memory + cpu->PC + 2);

            temp2 = *((uint8_t *)memory + temp);
            temp2 = temp2 << 8;
            temp2 |= *((uint8_t *)memory + temp + 1);
            temp2 = temp2 << 8;
            temp2 |= *((uint8_t *)memory + temp + 2);
```

```
395
396             if(cpu->A == temp2)
397             {
398                 cpu->SW = '=';
399             }
400             else if(cpu->A < temp2)
401             {
402                 cpu->SW = '<';
403             }
404             else
405             {
406                 cpu->SW = '>';
407             }
408
409             cpu->PC += 3;
410             break;
411
412         case 48:
413             printf("JEQ     ");
414             temp = cpu->PC;
415             if(cpu->SW == '=')
416             {
417                 cpu->PC = *((uint8_t *)memory + temp + 1);
418                 cpu->PC = cpu->PC  << 8;
419                 cpu->PC |= *((uint8_t *)memory + temp + 2);
420
421             }
422             else
423             {
424                 cpu->PC += 3;
425             }
426             break;
427
428         case 72:
429             printf("JSUB    ");
430             temp = cpu->PC;
431             cpu->L = cpu->PC + 3;
432             cpu->PC = *((uint8_t *)memory + temp + 1);
433             cpu->PC = cpu->PC  << 8;
434             cpu->PC |= *((uint8_t *)memory + temp + 2);
435             break;
436
437         case 8:
438             printf("LDL     ");
439             temp = *((uint8_t *)memory + cpu->PC + 1);
440             temp = temp << 8;
441             temp |= *((uint8_t *)memory + cpu->PC + 2);
442
443             if(*((uint8_t *)memory + temp) == '.')
444             {
445                 cpu->L = 0;
446             }
447             else
448             {
449                 cpu->L = *((uint8_t *)memory + temp);
450             }
451
452             cpu->L = cpu->L << 8;
453
454             if(*((uint8_t *)memory + temp + 1) == '.')
455             {
456                 cpu->L |= 0;
457             }
458             else
459             {
460                 cpu->L |= *((uint8_t *)memory + temp + 1);
461             }
462
463             cpu->L = cpu->L << 8;
464
465             if(*((uint8_t *)memory + temp + 2) == '.')
466             {
467                 cpu->L |= 0;
468             }
469             else
470             {
471                 cpu->L |= *((uint8_t *)memory + temp + 2);
472             }
473
```

```
                cpu->PC += 3;
                break;

            case 68:
                printf("OR      ");
                cpu->PC += 3;
                break;

            case 232:
                printf("STSW    ");
                cpu->PC += 3;
                break;

            case 224:
                printf("TD      ");
                temp = *((uint8_t *)memory + cpu->PC + 1);
                temp = temp << 8;
                temp |= *((uint8_t *)memory + cpu->PC + 2);


            temp2 = *((uint8_t *)memory + temp);
                //printf("%02X\n", temp2);

                if(temp2 == hex_to_dex("F1"))
                {
                    if(input != NULL)
                    {
                        cpu->SW = '<';
                    }
                    else
                    {
                        cpu->SW = '=';
                    }
                }

                if(temp2 == hex_to_dex("05"))
                {
                    if(output != NULL)
                    {
                        cpu->SW = '<';
                    }
                    else
                    {
                        cpu->SW = '=';
                    }
                }

                /*
                if(*((uint8_t *)memory + temp) == '.')
                {
                    cpu->SW = '=';
                }
                else
                {
                    cpu->SW = '<';
                }
                */

                cpu->PC += 3;
                break;

            default:
                break;
        }
        printf("A : %06X  X : %06X  L : %06X  PC : %06X  SW : %06X\n\n", cpu->A, cpu-
>X, cpu->L, cpu->PC, cpu->SW);
    }
    fclose(input);
    fclose(output);

    printf("\n");
}
```