PROGRAM CODE

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

char* intToHex(int num) {
	char* buff = malloc(7 * sizeof(char));
	sprintf(buff, "%X", num);
	return buff;
}

int hexToInt(char* buff) {
	int num;
	sscanf(buff, "%X", &num);
	return num;
}

void displayObj(char* filename) {
	FILE* fp = fopen(filename, "r");

	if(!fp) {
		printf("No such file: %s\n", filename);
		exit(0);
	}

	char buff[100];

	printf("\nObject program:\n");
	while(1) {
		fgets(buff, 100, fp);
		if(feof(fp))
			break;
		fputs(buff, stdout);
	}
	printf("\n");

	fclose(fp);
}

void displayMem() {
	FILE* memFile = fopen("memory", "r");

	char buff[10];
	fgets(buff, 10, memFile);

	printf("Memory Allocation:\n");

	while(!feof(memFile)) {
		printf("%s", buff);

		fgets(buff, 10, memFile);
	}
}

void load(char* filename) {
	FILE* objFile = fopen(filename, "r");
	FILE* memFile = fopen("memory", "w");
	char pgName[6];
	int pgAddr;
	int pgLength;
```

```
char buff[100];
fgets(buff, 100, objFile);

if(buff[0] == 'H') {
	for(int i = 2; i < 8; i++) {
		pgName[i-2] = buff[i];

		if(buff[i+1] == ' ') {
			pgName[i-1] = '\0';
			break;
		}
	}

	printf("Program Name: %s\n", pgName);

	char temp[] = {buff[9], buff[10], buff[11], buff[12], buff[13], buff[14], '\0'};

	pgAddr = hexToInt(temp);

	printf("Program Start Address: %XH\n", pgAddr);

	for(int i = 0; i < 6; i++)
		temp[i] = buff[i+16];

	pgLength = hexToInt(temp);

	printf("Program Length: %XH\n", pgLength);
} else {
	printf("Invalid object program: %s\n", filename);
	fclose(objFile);
	fclose(memFile);
	return;
}

fgets(buff, 100, objFile);

int addr = pgAddr, textLen;

while(buff[0] != 'E') {
	char* token = strtok(buff, "^");

	token = strtok(NULL, "^");

	while(addr != hexToInt(token)) {
		fputs(intToHex(addr), memFile);
		fputs(": XX\n", memFile);
		addr++;
	}

	addr = hexToInt(token);

	token = strtok(NULL, "^");

	textLen = hexToInt(token);

	while(1) {
		token = strtok(NULL, "^");
		if(token == NULL)
			break;
		else {
			if (token[strlen(token) - 1] == '\n')
				token[strlen(token) - 1] = '\0';
```

```
                for(int i = 0; i < strlen(token); i = i + 2) {
                    fputs(intToHex(addr), memFile);
                    fputs(": ", memFile);

                    char temp[] = {token[i], token[i+1], '\n', '\0'};
                    fputs(temp, memFile);

                    addr++;
                }
            }
        }

        fgets(buff, 100, objFile);
    }

    while(addr <= (pgAddr + pgLength)) {
        fputs(intToHex(addr), memFile);
        fputs(": XX\n", memFile);
        addr++;
    }

    char temp[] = {buff[2], buff[3], buff[4], buff[5], buff[6], buff[7], '\0'};

    int execAddr = atoi(temp);

    printf("Program Execution Address: %dH\n", execAddr);

    fclose(objFile);
    fclose(memFile);
    displayMem();
}

void main() {
    char* filename = malloc(20 * sizeof(char));

    printf("Enter filename of object program: ");
    fgets(filename, 20, stdin);

    for(int i = 0; i < 20; i++)
        if(filename[i] == '\n')
            filename[i] = '\0';

    displayObj(filename);
    load(filename);
}
```

OUTPUT

```
Enter filename of object program: obj

Object program:
H^PG1   ^001000^000072
T^001000^1E^041042^10105D^10106F^00904E^40104B^28104B^301018^3C102D^04105D^00904E
T^00101E^1E^04106F^0C906C^00106F^181045^0C106F^00105D^181045^0C105D^04105D^281048
T^00103C^12^381009^4C0000^000000^000003^00000F^800000
E^001000

Program Name: PG1
```

Program Start Address: 1000H
Program Length: 72H
Program Execution Address: 1000H
Memory Allocation:

```
1000: 04
1001: 10
1002: 42
1003: 10
1004: 10
1005: 5D
1006: 10
1007: 10
1008: 6F
1009: 00
100A: 90
100B: 4E
100C: 40
100D: 10
100E: 4B
100F: 28
1010: 10
1011: 4B
1012: 30
1013: 10
1014: 18
1015: 3C
1016: 10
1017: 2D
1018: 04
1019: 10
101A: 5D
101B: 00
101C: 90
101D: 4E
101E: 04
101F: 10
1020: 6F
1021: 0C
1022: 90
1023: 6C
1024: 00
1025: 10
1026: 6F
1027: 18
1028: 10
1029: 45
102A: 0C
102B: 10
102C: 6F
102D: 00
102E: 10
102F: 5D
1030: 18
1031: 10
1032: 45
1033: 0C
1034: 10
1035: 5D
1036: 04
1037: 10
1038: 5D
1039: 28
103A: 10
```

```
103B: 48
103C: 38
103D: 10
103E: 09
103F: 4C
1040: 00
1041: 00
1042: 00
1043: 00
1044: 00
1045: 00
1046: 00
1047: 03
1048: 00
1049: 00
104A: 0F
104B: 80
104C: 00
104D: 00
104E: XX
104F: XX
1050: XX
1051: XX
1052: XX
1053: XX
1054: XX
1055: XX
1056: XX
1057: XX
1058: XX
1059: XX
105A: XX
105B: XX
105C: XX
105D: XX
105E: XX
105F: XX
1060: XX
1061: XX
1062: XX
1063: XX
1064: XX
1065: XX
1066: XX
1067: XX
1068: XX
1069: XX
106A: XX
106B: XX
106C: XX
106D: XX
106E: XX
106F: XX
1070: XX
1071: XX
1072: XX
```