

```

#include <stdlib.h>
#include <strings.h>
#include <fstream.h>
#include <iostream.h>
#include "Table.h"
#include "Loader.h"
#include "Loader2.h"

void main(int argc, char* argv[], char* envp[])
{
    if (argc < 4)
    {
        cerr << "Usage: Loader [assembled_source_file(s)] <object file> initial_load_address\n";
        exit(1);
    }

    Table ES_Table(50);
    ifstream* inputs = new ifstream[argc-3];
    int File_Number = 0;
    while (File_Number < (argc - 3))
    {
        inputs[File_Number].open(argv[File_Number+1]);
        if (! inputs[File_Number].is_open())
        {
            cerr << "Incorrect filename \"" << argv[File_Number+1] << "\"\n";
            exit(1);
        }
        if (inputs[File_Number].eof())
        {
            cerr << "File \"" << argv[File_Number+1] << "\" empty or permissions set
incorrectly\n";
            exit(1);
        }
        File_Number++;
    }

    int Total_Length = 0, Begin_Execution = 0;

    Total_Length = Loader_One(ES_Table, inputs, argc, Begin_Execution);
    if (Total_Length > 255)
    {
        cerr << "Program too large to fit into memory\n";
        exit(1);
    }
    ofstream object_out;
    object_out.open(argv[argc-2]);
    if (! object_out.is_open())
    {
        cerr << "Output file \"" << argv[argc-2] << "\" is read-only or file permissions set
incorrectly\n";
        exit(1);
    }
    if (argv[argc-1][0] != '1' && argv[argc-1][0] != '2' && argv[argc-1][0] != '3' && argv[argc-1][0] != '4' && argv[argc-1][0] != '5' && argv[argc-1][0] != '6' && argv[argc-1][0] != '7' && argv[argc-1][0] != '8' && argv[argc-1][0] != '9')
    {
        cerr << "IPLA \"" << argv[argc-1] << "\" not supplied, or not in decimal format\n";
        exit(1);
    }
    int IPLA = atoi(argv[argc-1]);
    if ((IPLA + Total_Length) > 255)
    {
        cerr << "Not enough memory to start at address " << IPLA << "\n";
        exit(1);
    }
    ifstream middle;
    middle.open("intermediate");

```

```
Loader_Two(ES_Table, IPLA, Begin_Execution, Total_Length, inputs, object_out, middle, ↵  
argc);
```

```
cout << "Program loaded successfully\n";  
exit(1);
```

```
}
```