```
void Loader_Two(Table ES_Table, int IPLA, int Begin_Execution, int Total_Length, ifstream*↙
    & inputs, ofstream& outs, ifstream& middle, int number_of_files)
{
    char init[3];
    middle >> init;
    int PLA = (atoi(init)+IPLA);
    outs << 'H';
    outs.width(2);
    outs.fill('0');
    outs << hex << IPLA + Begin_Execution;
    outs << "ODProg";
    outs.width(2);
    outs.fill('0');
    outs << hex << IPLA;
    outs.width(2);
    outs.fill('0');
    outs << hex << Total_Length << '\n';
    int File_Number = 0;

    while (File_Number < (number_of_files - 3))
    {
    while (!inputs[File_Number].eof())
    {
        char buffer[80];
        char* token;
        inputs[File_Number].getline(buffer, 80);
        if (strlen(buffer) > 0)
        {
        token = strtok(buffer, "."); //T
        outs << 'T';
        token = strtok(NULL, "."); //address
        outs.width(2);
        outs.fill('0');
        outs << hex << PLA + atoi(token);

        token = strtok(NULL, "."); //data
        int data = atoi(token);
        token = strtok(NULL, "."); //M/X/NULL
        if (token == NULL)
        {
            //do nothing
        }
        else if (token[0] == 'M')
        {
            data += PLA;
        }
        else if (token[0] == 'X')
        {
            token = strtok(NULL, ".");
            if (token == NULL)
            {
            cerr << "Missing EXTernal symbol or incorrect source format\n";
            exit(1);
            }
            else if (!ES_Table.Is_In_Table(token))
            {
            cerr << "EXTernal symbol \"" << token << "\" not defined\n";
            exit(1);
            }
            else
            {
            data += (IPLA + ES_Table.Get_Value(token));
            }
        }
        outs.width(5);
        outs.fill('0');
        outs << hex << data << '\n';
        }
```

```
    }
    File_Number++;
    middle >> init;
    PLA = (atoi(init) + IPLA);
    }
}
```