Name: Mukul Rai

Student ID: 700748568

GitHub Link for Project: https://github.com/raimukul/Malware_Project

Project 1

You should parse the attached two applications for their DOS headers and NT headers (including signature, coff header, and optional header) and print out their values in a format like (fields name: field value). The codes should be written in C or C++. You can also download the applications at https://github.com/squarekyzhong/project1-3.git
For example, Dos Header has 19 fields; you print like
e_magic: 5A4D
e_cblp: 90
e_cp: 3
e_crlc : 0
e_cparhdr: 4
…
Similarly, for NT headers.
All values should be in hexadecimal.
You should also download the Firefox browser to download the applications because Google Chrome will check them as malicious. https://www.mozilla.org/en-US/firefox/new/

Code (Using C Programming Language)

```
#include <stdio.h>
#include <windows.h>
#define REDFONT "\x1B[31m"
#define GRNFONT "\x1B[32m"

int main(int argc, char *argv[])
{

    if (argc != 2)
    {
        printf("Usage: %s <filename> \n \n", argv[0]);
        return 1;
    }

    FILE *file = fopen(argv[1], "rb");
    if (file == NULL)
    {
        printf("Error: Unable to open application you provided '%s'\n", argv[1]);
        return 1;
    }
```

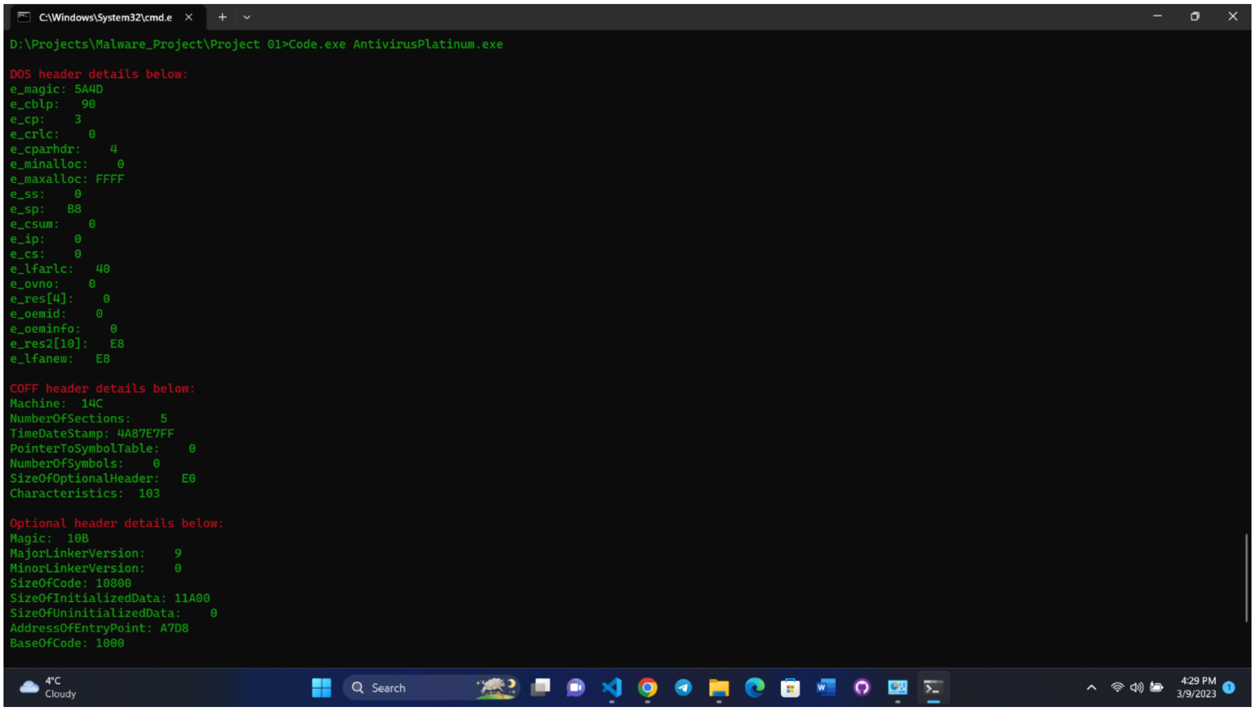
```c
// This below code will read the DOS header of an application
IMAGE_DOS_HEADER dos_header;
fread(&dos_header, sizeof(dos_header), 1, file);

// This below code will verify the DOS signature of an application
if (dos_header.e_magic != IMAGE_DOS_SIGNATURE)
{
    printf("Error: Invalid DOS signature of application \n");
    fclose(file);
    return 1;
}

// This below code will print out the DOS header fields of your application
printf("\n" REDFONT);
printf("DOS header details below: \n" GRNFONT);
printf("e_magic: % 4X\n", dos_header.e_magic);
printf("e_cblp: % 4X\n", dos_header.e_cblp);
printf("e_cp: % 4X\n", dos_header.e_cp);
printf("e_crlc: % 4X\n", dos_header.e_crlc);
printf("e_cparhdr: % 4X\n", dos_header.e_cparhdr);
printf("e_minalloc: % 4X\n", dos_header.e_minalloc);
printf("e_maxalloc: % 4X\n", dos_header.e_maxalloc);
printf("e_ss: % 4X\n", dos_header.e_ss);
printf("e_sp: % 4X\n", dos_header.e_sp);
printf("e_csum: % 4X\n", dos_header.e_csum);
printf("e_ip: % 4X\n", dos_header.e_ip);
printf("e_cs: % 4X\n", dos_header.e_cs);
printf("e_lfarlc: % 4X\n", dos_header.e_lfarlc);
printf("e_ovno: % 4X\n", dos_header.e_ovno);
printf("e_res[4]: % 4X\n", dos_header.e_res[4]);
printf("e_oemid: % 4X\n", dos_header.e_oemid);
printf("e_oeminfo: % 4X\n", dos_header.e_oeminfo);
printf("e_res2[10]: % 4X\n", dos_header.e_res2[10]);
printf("e_lfanew: % 4X\n", dos_header.e_lfanew);
```

```
// This below code will seek to the NT header offset of an application
fseek(file, dos_header.e_lfanew, SEEK_SET);

// This below code will read the NT header signature of an application
DWORD nt_signature;
fread(&nt_signature, sizeof(nt_signature), 1, file);

// This below code will verify the NT signature of an application
if (nt_signature != IMAGE_NT_SIGNATURE)
{
    printf("Error: Invalid NT signature\n");
    fclose(file);
    return 1;
}

// This below code will read the COFF header of an application
IMAGE_FILE_HEADER coff_header;
fread(&coff_header, sizeof(coff_header), 1, file);

// This below code will print out the COFF header fields of your application
printf("\n" REDFONT);
printf("COFF header details below:\n" GRNFONT);
printf("Machine: % 4X\n", coff_header.Machine);
printf("NumberOfSections: % 4X\n", coff_header.NumberOfSections);
printf("TimeDateStamp: % 4X\n", coff_header.TimeDateStamp);
printf("PointerToSymbolTable: % 4X\n", coff_header.PointerToSymbolTable);
printf("NumberOfSymbols: % 4X\n", coff_header.NumberOfSymbols);
printf("SizeOfOptionalHeader: % 4X\n", coff_header.SizeOfOptionalHeader);
printf("Characteristics: % 4X\n", coff_header.Characteristics);

// This below code will read the optional header
IMAGE_OPTIONAL_HEADER optional_header;
fread(&optional_header, sizeof(optional_header), 1, file);

// This below code will print out the optional header fields
printf("\n" REDFONT);
```

```
    printf("Optional header details below:\n" GRNFONT);
    printf("Magic: % 4X\n", optional_header.Magic);
    printf("MajorLinkerVersion: % 4X\n", optional_header.MajorLinkerVersion);
    printf("MinorLinkerVersion: % 4X\n", optional_header.MinorLinkerVersion);
    printf("SizeOfCode: % 4X\n", optional_header.SizeOfCode);
    printf("SizeOfInitializedData: % 4X\n", optional_header.SizeOfInitializedData);
    printf("SizeOfUninitializedData: % 4X\n", optional_header.SizeOfUninitializedData);
    printf("AddressOfEntryPoint: % 4X\n", optional_header.AddressOfEntryPoint);
    printf("BaseOfCode: % 4X\n", optional_header.BaseOfCode);
}
```

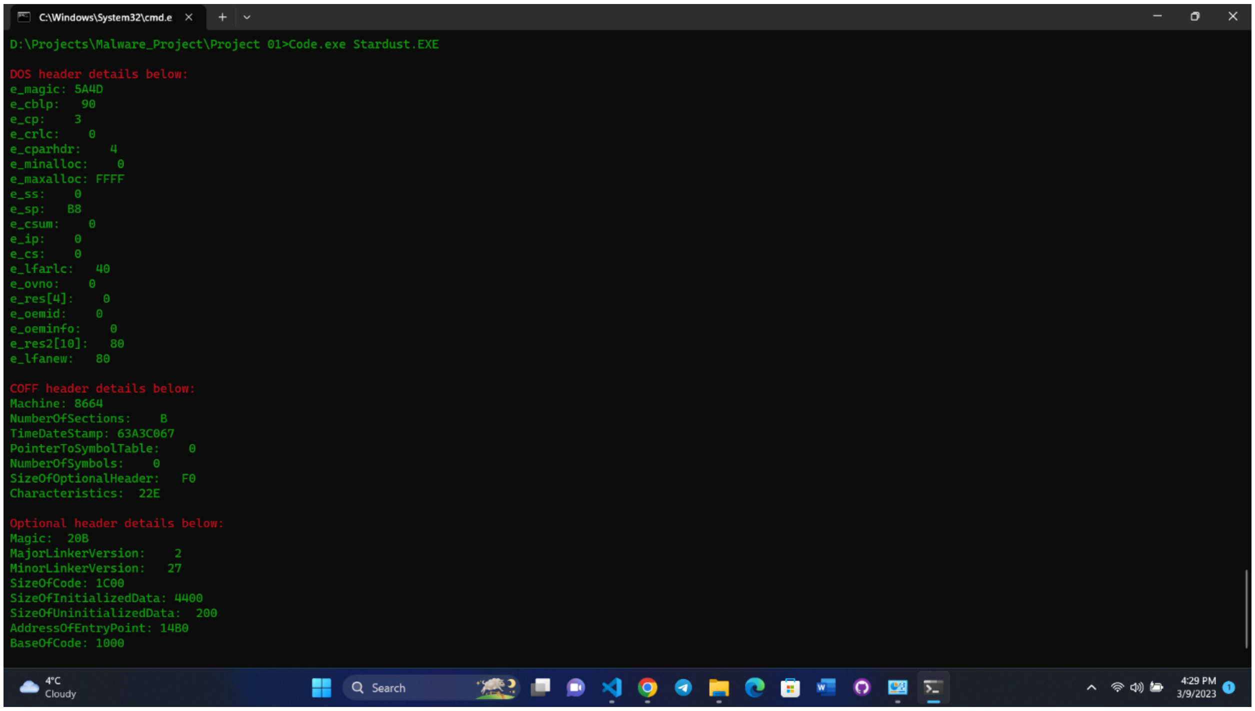
Output

1. *Output for AntivirusPlatinum.exe*



```
D:\Projects\Malware_Project\Project 01>Code.exe AntivirusPlatinum.exe

DOS header details below:
e_magic: 5A4D
e_cblp:   90
e_cp:    3
e_crlc:    0
e_cparhdr:    4
e_minalloc:    0
e_maxalloc: FFFF
e_ss:    0
e_sp:   B8
e_csum:    0
e_ip:    0
e_cs:    0
e_lfarlc:   40
e_ovno:    0
e_res[4]:    0
e_oemid:    0
e_oeminfo:    0
e_res2[10]:   E8
e_lfanew:   E8

COFF header details below:
Machine:  14C
NumberOfSections:    5
TimeDateStamp: 4A87E7FF
PointerToSymbolTable:    0
NumberOfSymbols:    0
SizeOfOptionalHeader:   E0
Characteristics:  103

Optional header details below:
Magic:  10B
MajorLinkerVersion:    9
MinorLinkerVersion:    0
SizeOfCode: 10800
SizeOfInitializedData: 11A00
SizeOfUninitializedData:    0
AddressOfEntryPoint: A7D8
BaseOfCode: 1000
```

2. *Output For Stardust.exe*



```
D:\Projects\Malware_Project\Project 01>Code.exe Stardust.EXE

DOS header details below:
e_magic: 5A4D
e_cblp:   90
e_cp:    3
e_crlc:    0
e_cparhdr:    4
e_minalloc:    0
e_maxalloc: FFFF
e_ss:    0
e_sp:   B8
e_csum:    0
e_ip:    0
e_cs:    0
e_lfarlc:   40
e_ovno:    0
e_res[4]:    0
e_oemid:    0
e_oeminfo:    0
e_res2[10]:   80
e_lfanew:   80

COFF header details below:
Machine: 8664
NumberOfSections:    B
TimeDateStamp: 63A3C067
PointerToSymbolTable:    0
NumberOfSymbols:    0
SizeOfOptionalHeader:   F0
Characteristics:  22E

Optional header details below:
Magic:  20B
MajorLinkerVersion:    2
MinorLinkerVersion:   27
SizeOfCode: 1C00
SizeOfInitializedData: 4400
SizeOfUninitializedData:  200
AddressOfEntryPoint: 14B0
BaseOfCode: 1000
```