REVERESE ENGINEERING

Presented By:
BASHETTY ARUN KUMAR
EPITA-2018
Computer Security

Questions for Given exercise:

1. Which compiler was used?

2. When was this application compiled?

3. What is its SHA-1 hash value?

4. Which CPU platform is it compiled for? 32- or 64-bit versions?

5. What it its entry point? In which section is it?

6. What are the sections in the application?

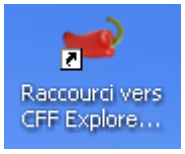
7. Is the entire application packed with UPX?

8. Is there any compressed/packed section?

9. What are the imported libraries? Give 1 API per imported library.

Applications that we are used to finding the above questions and patching the password is:

1. *CFF Explorer*



2. HxD

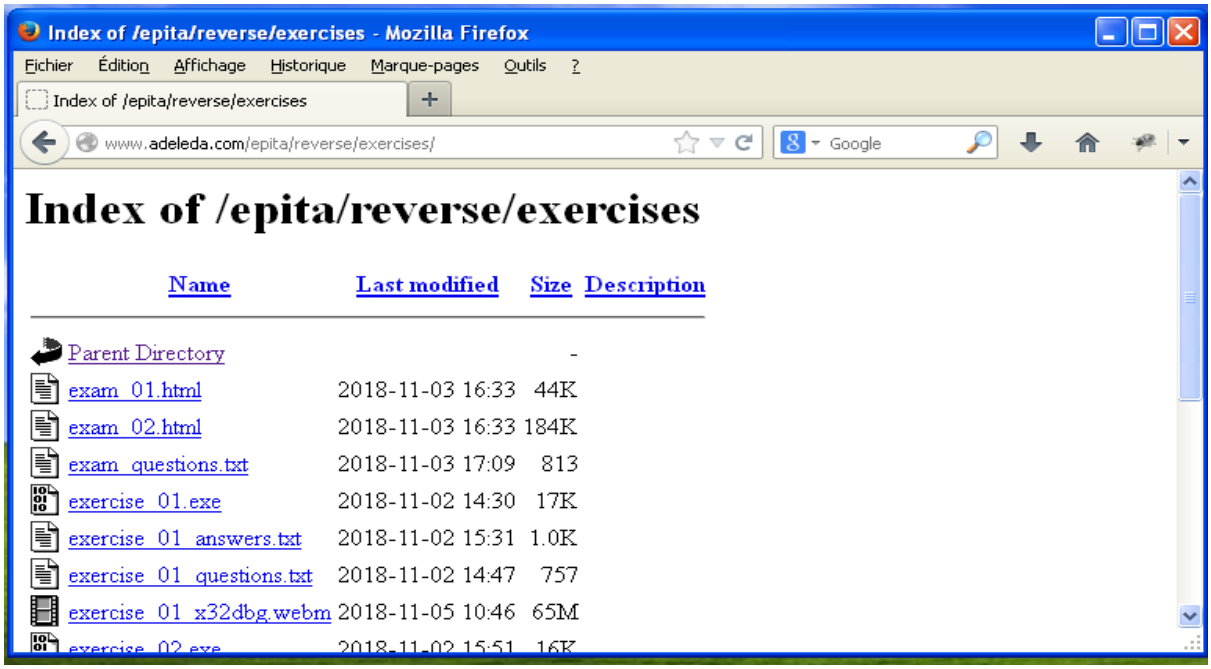
3.IDA Pro free

4.x32 Debugger

Steps for changing the file extension of given file =>

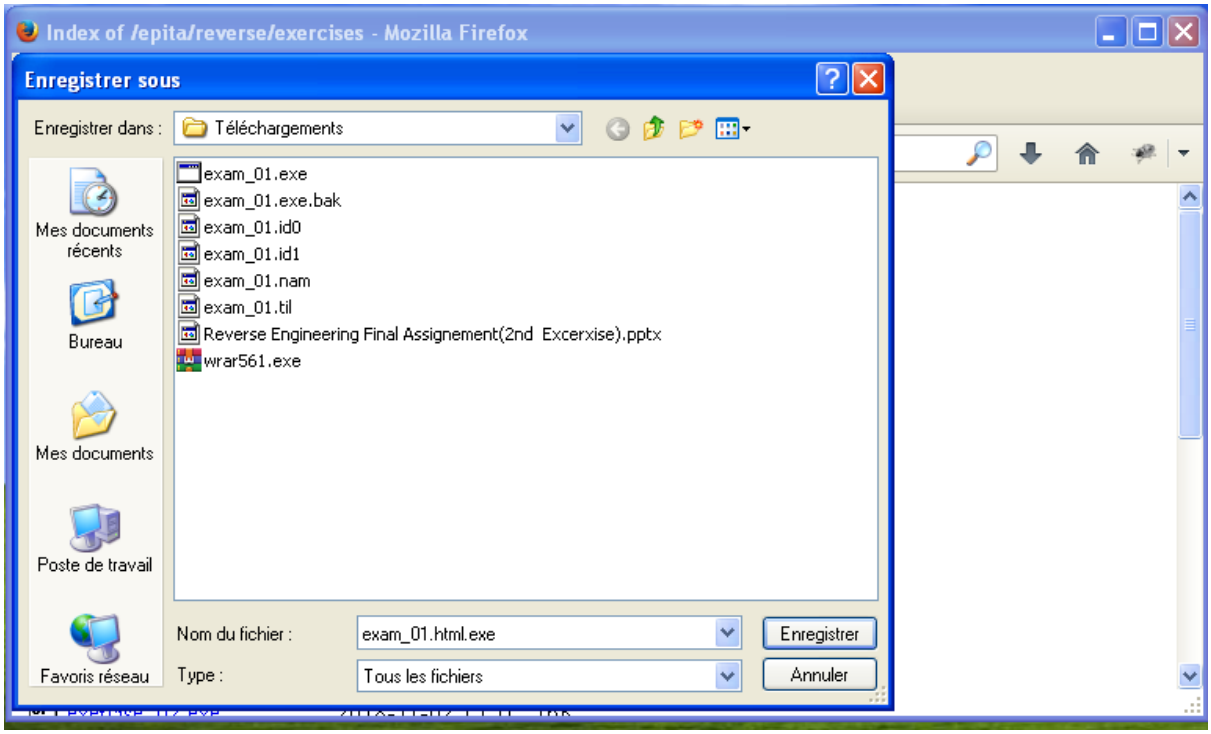
First, click on =>

www.adeleda.com/epita/reverse/excercises, it will go to **adeleda** web page and we can check the exam 01.html file.



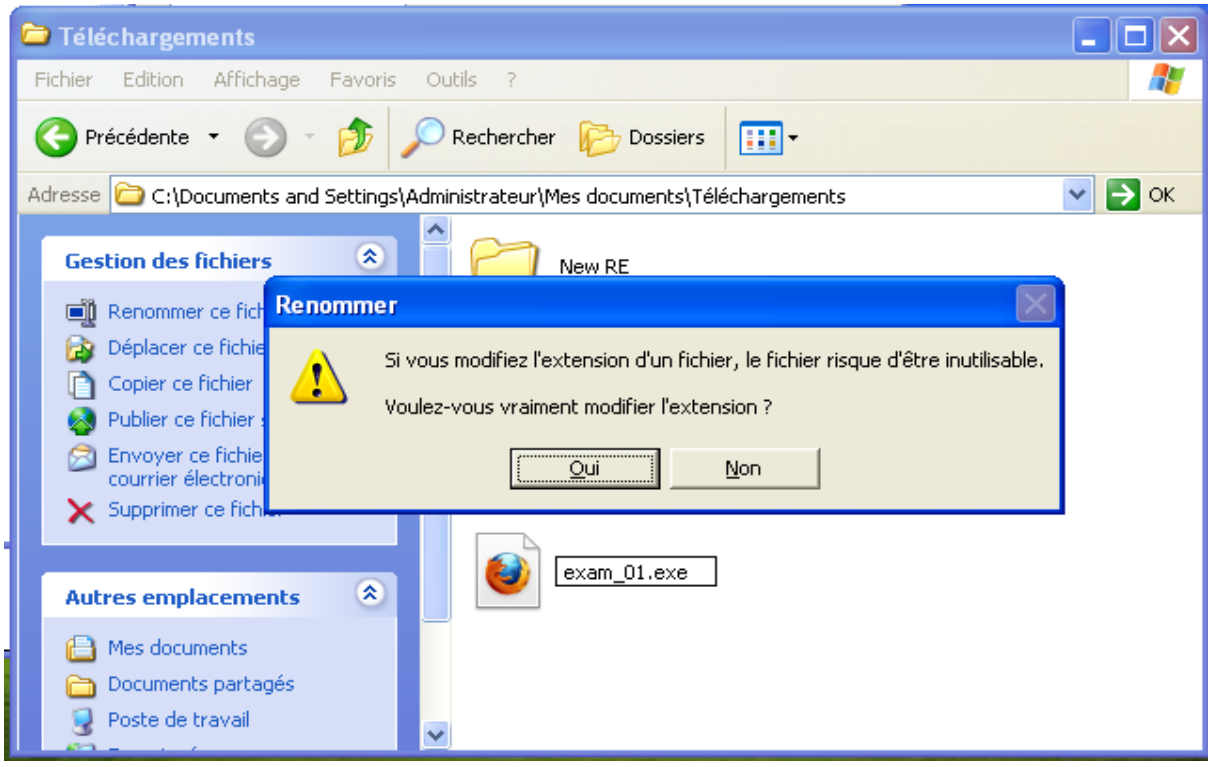
Right click on exam 01.html, you can see the list of files in below screenshot,

In the list, if you click on Save the file (Enregister la cible du lien sous.), you must save the extension by giving .exe and save it.



After saving the file by giving extension, you can see the file saved as both html and exe extension,

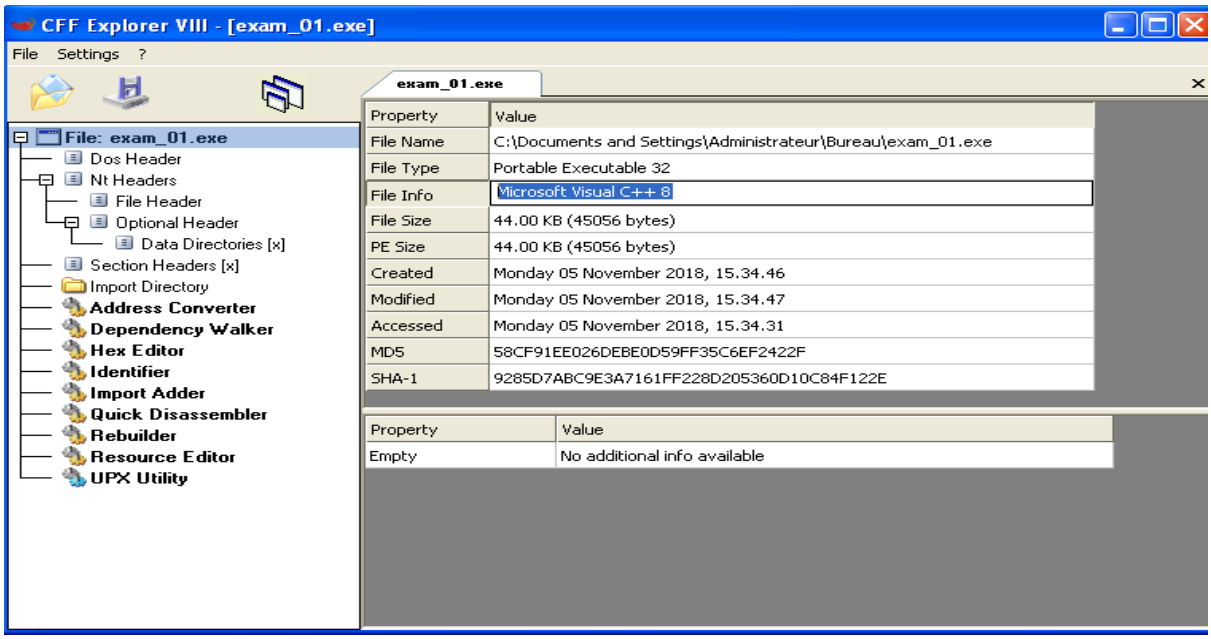
you change the extension to .exe, it will be shown the below screenshot and click on Oui.



1. Which compiler was used?

First import the exercise file in CFF explorer, when you click on File: exam1.exe, you can see Compiler details.

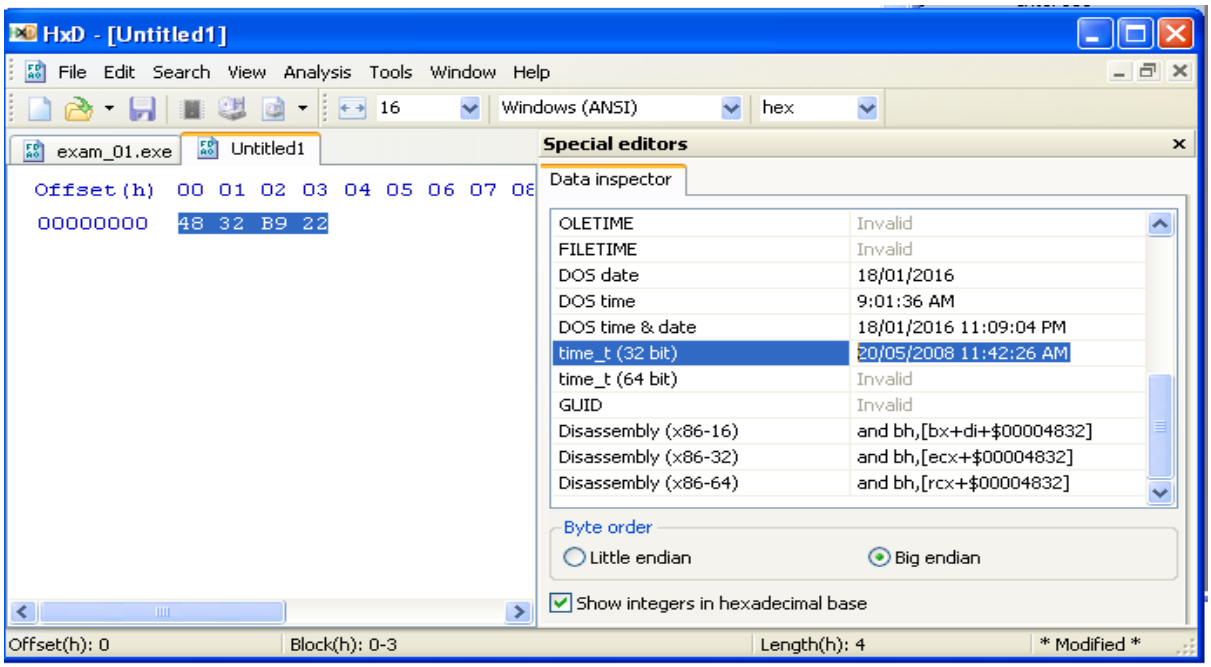
Compiler: **Microsoft Visual C++ 8**



(1st Step).

2. When was this application compiled?

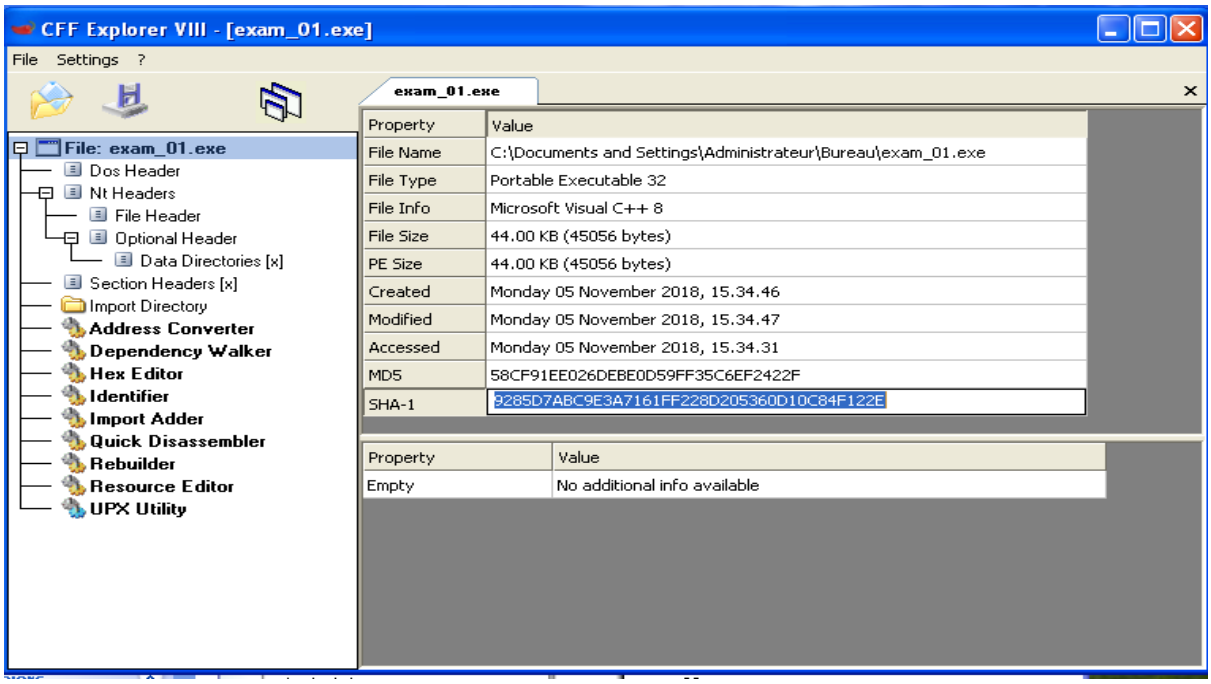
When you click on File=>Nt Header=>File Header, you can see TimeDateStamp with the value **4D CF 89 81.**



What is its SHA-1 hash value?

After Importing the file in CFF Explorer in the File: exam_01.exe, you can see sha1 value.

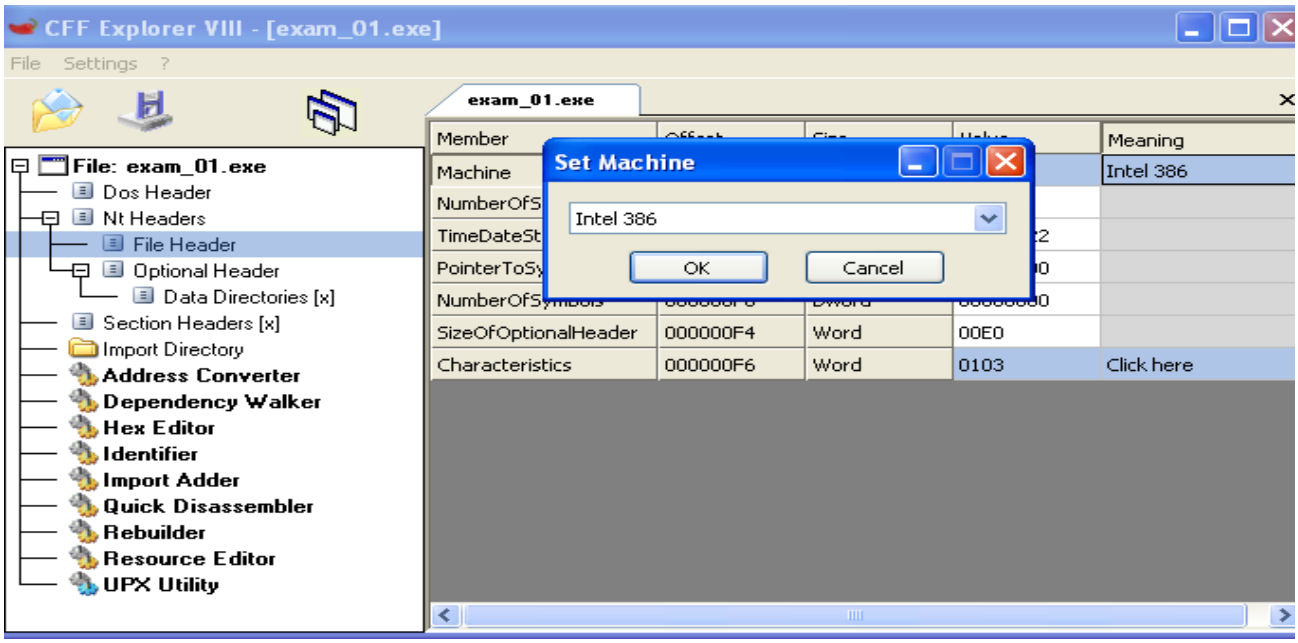
=>File=>Sha-1=> **9285D7ABC9E3A7161FF228D205360D10C84F122E**



4.Which CPU platform is it compiled for? 32- or 64-bit versions?

You can search 2 ways for CPU Platform compiled version

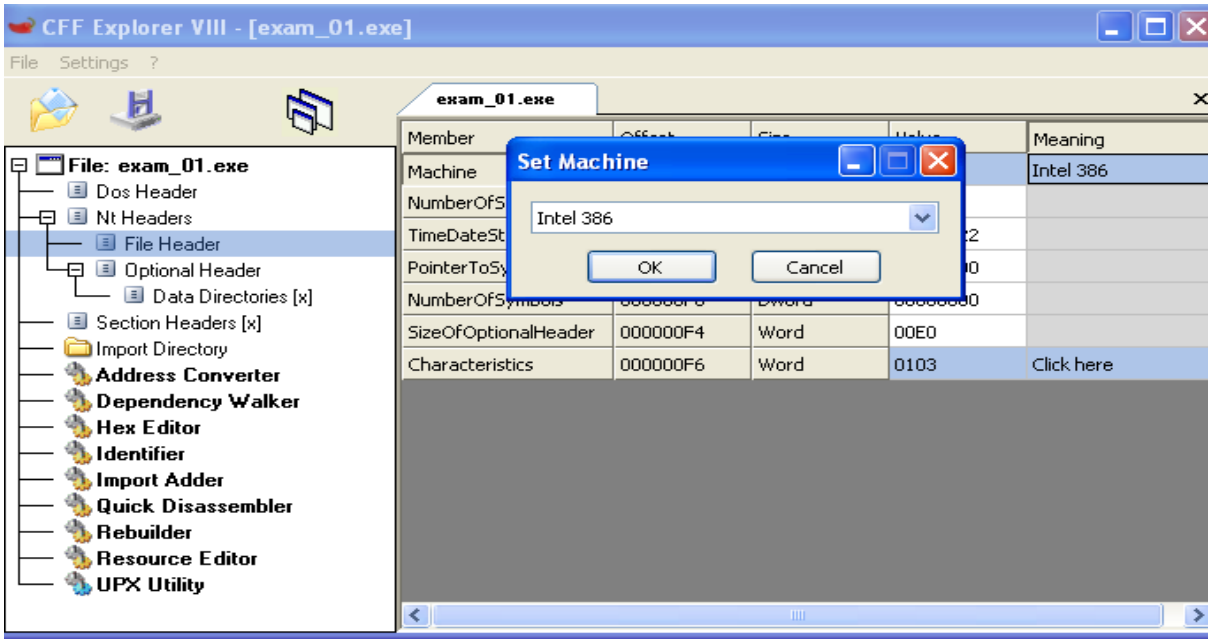
1st way: Click on Nt Folder, in File Header you can check the set machine displayed by box.



2nd Way:

To Check the CPU Platform compiled Version:

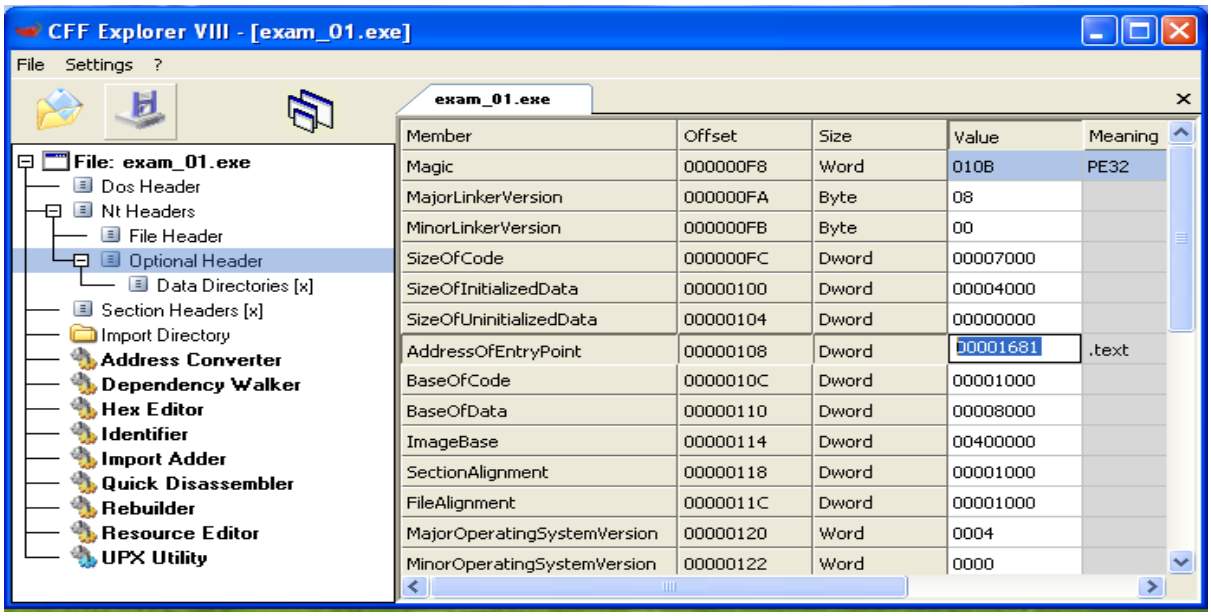
In the same File Header, If you click on Characteristics=>Click here, it will show dialog clicking on 32 bit.



5. What it its entry point? In which section is it?

Click on Nt Header, inside Optional Header yon can see **the AddressOfEntryPoint** with the value

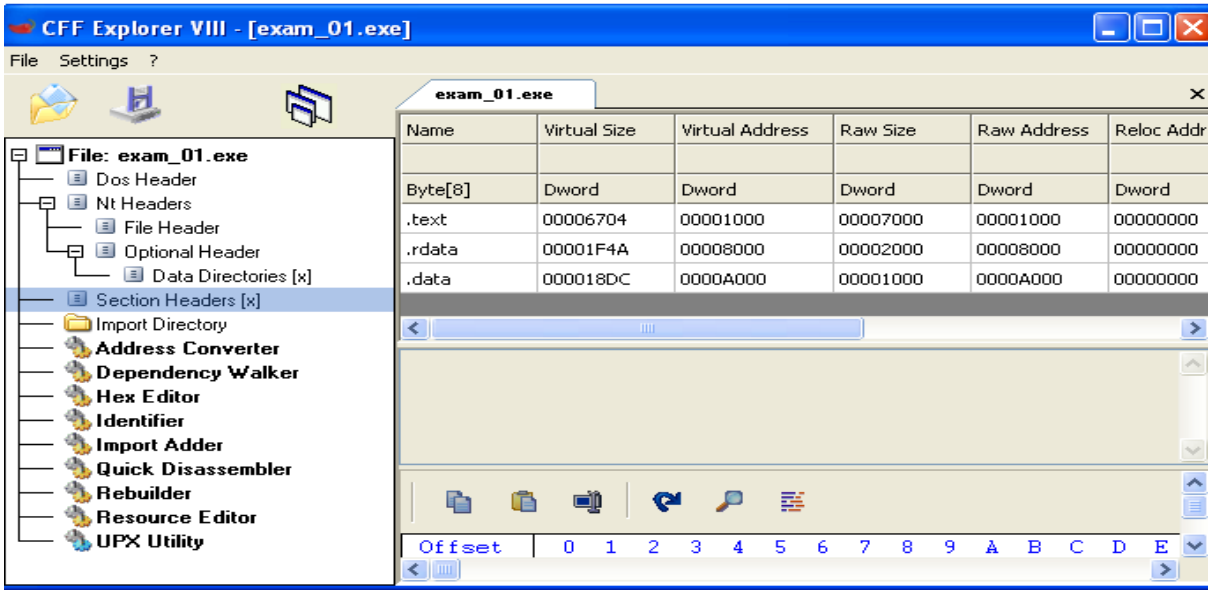
=> File=>Nt Headers=>Optional Header=>AddressOfEntryPoint=>1681 in .text section



6. What are the sections in the application?

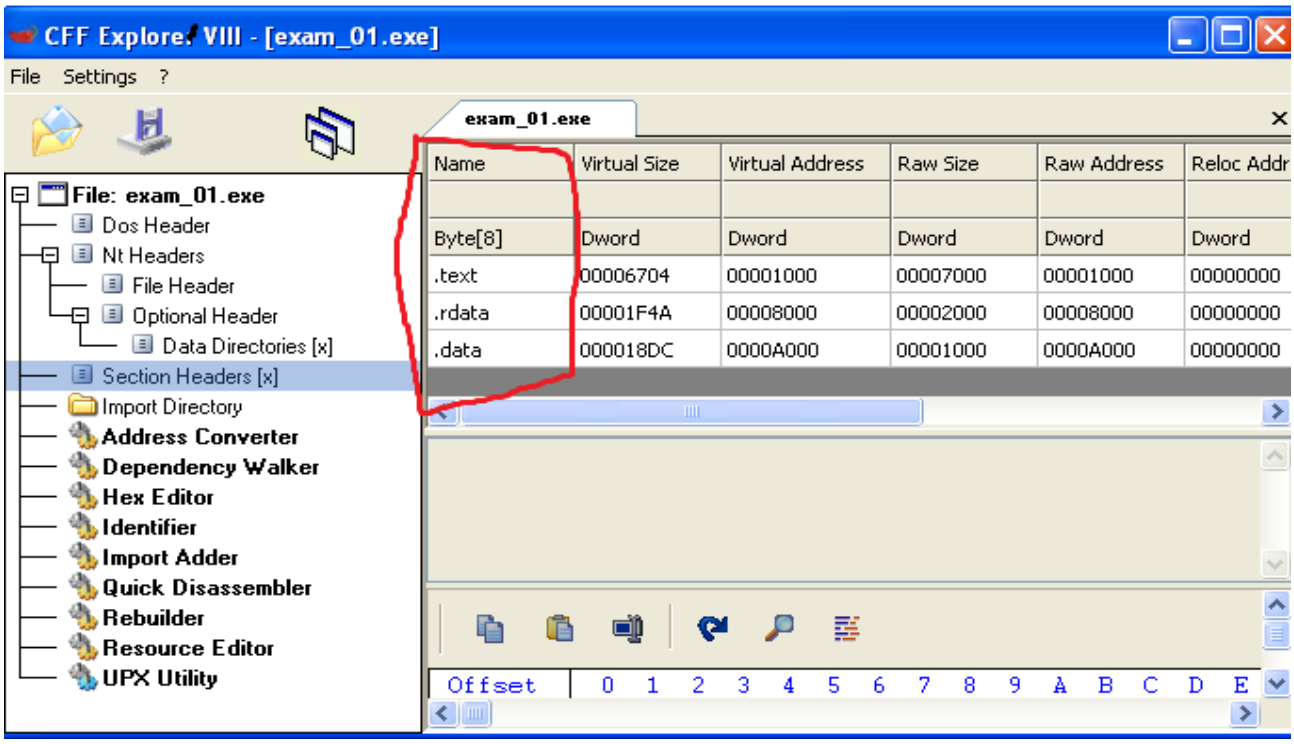
If you click on Section Header[x], you can see the Byte [8] with Sections.

=>File=>Section Headers=>5 (. text,. rdata, .data)



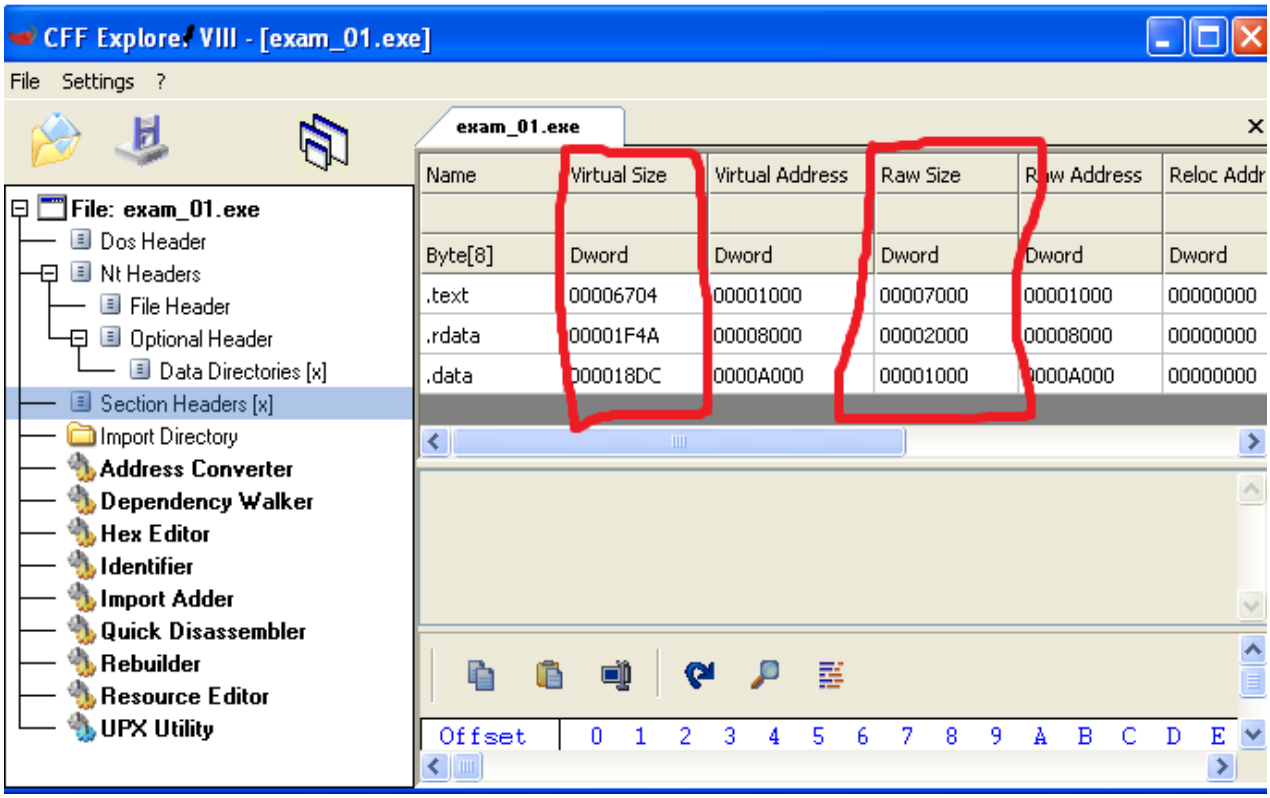
7. Is the entire application packed with UPX ?

No. Because the sections are not names .UPX0 or .UPX1



8. Is there any compressed/packed section?

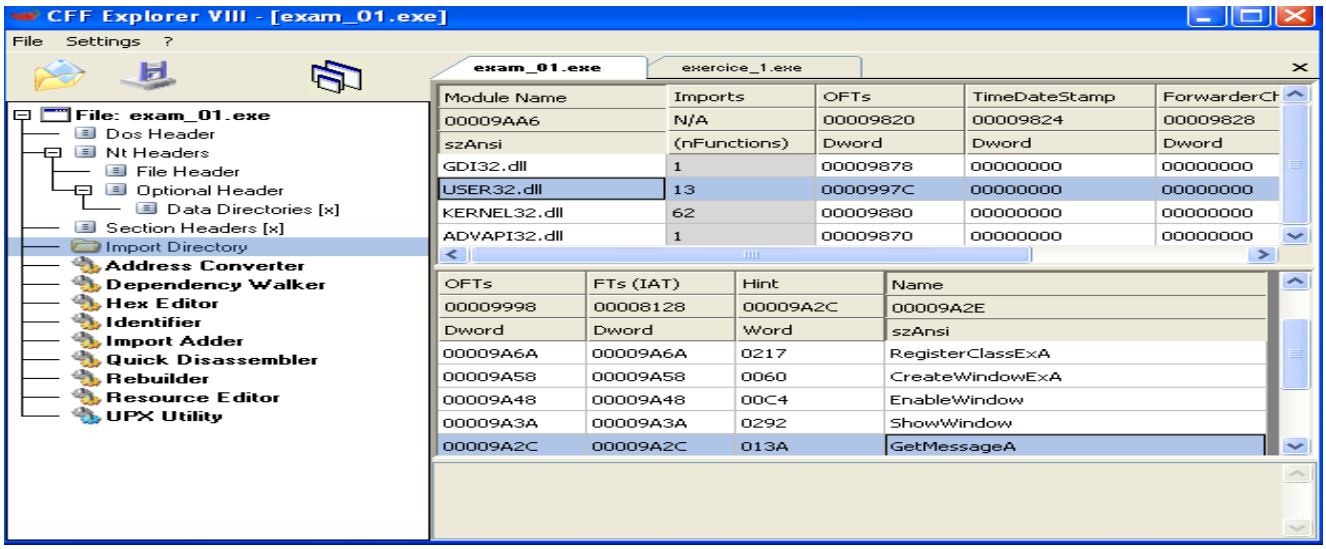
=> no, the Raw size and Virtual size of each section are almost equal



9. What are the imported libraries? Give 1 API per imported library.

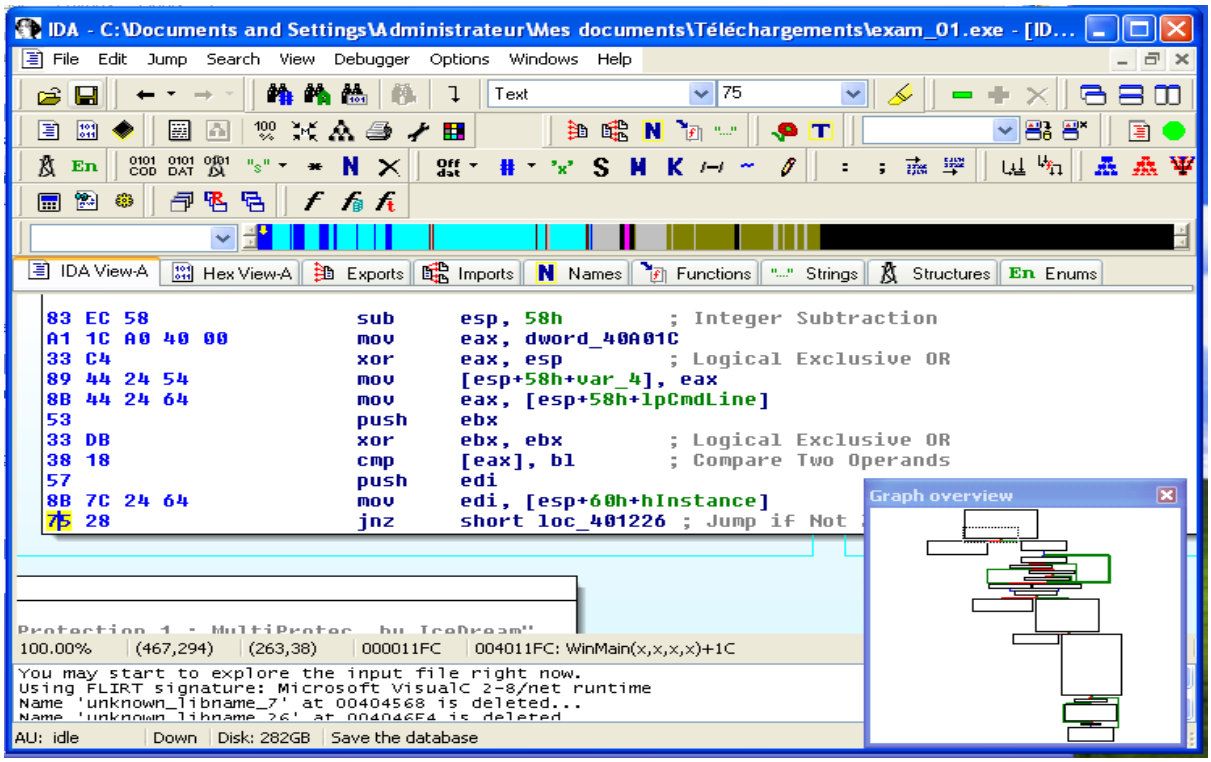
=> File=>Import Directory=> GDI32.dll, USER32.dll, Kernel32.dll, ADVAPI32.dll

(**ExitProcess** in Kernal132.dll, **CreateSolidBrush** in GDI32.dll, **RegOpenKeyExA** in ADVAPI32.dll, **RegisterClassExA** in USER32.dll)

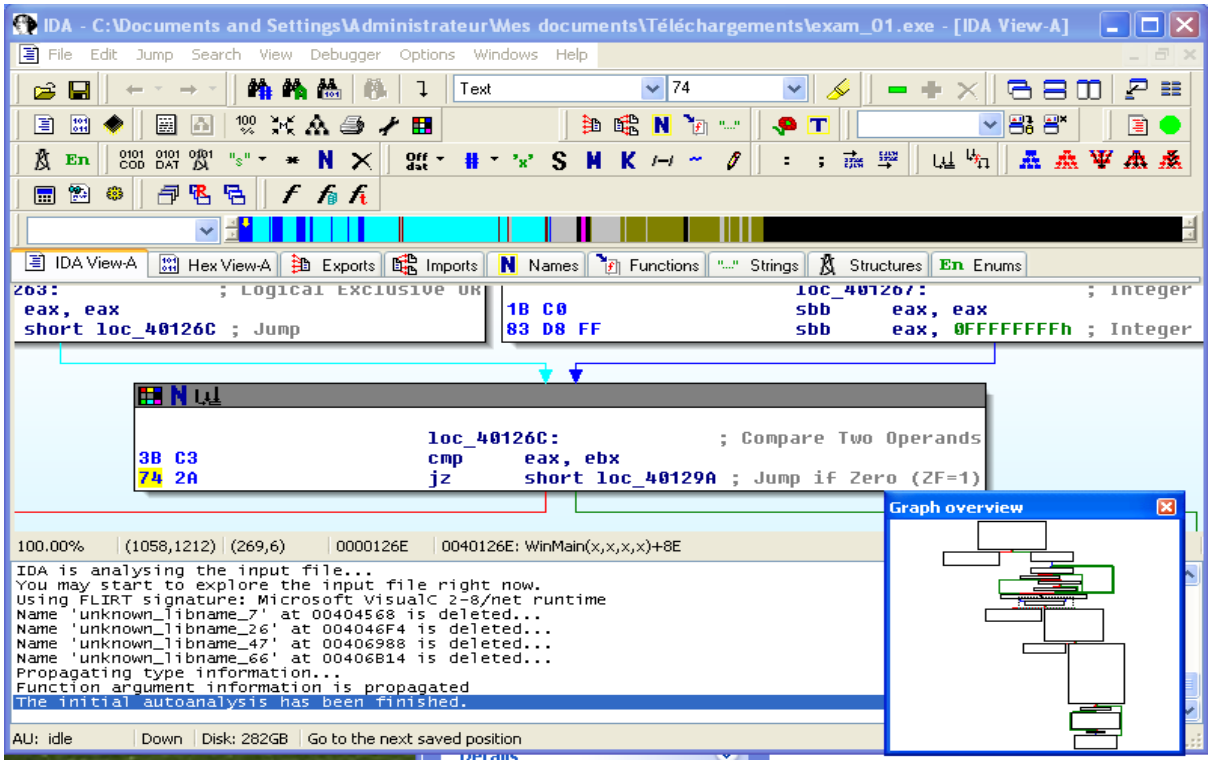


Process for patching the password by using below Steps:

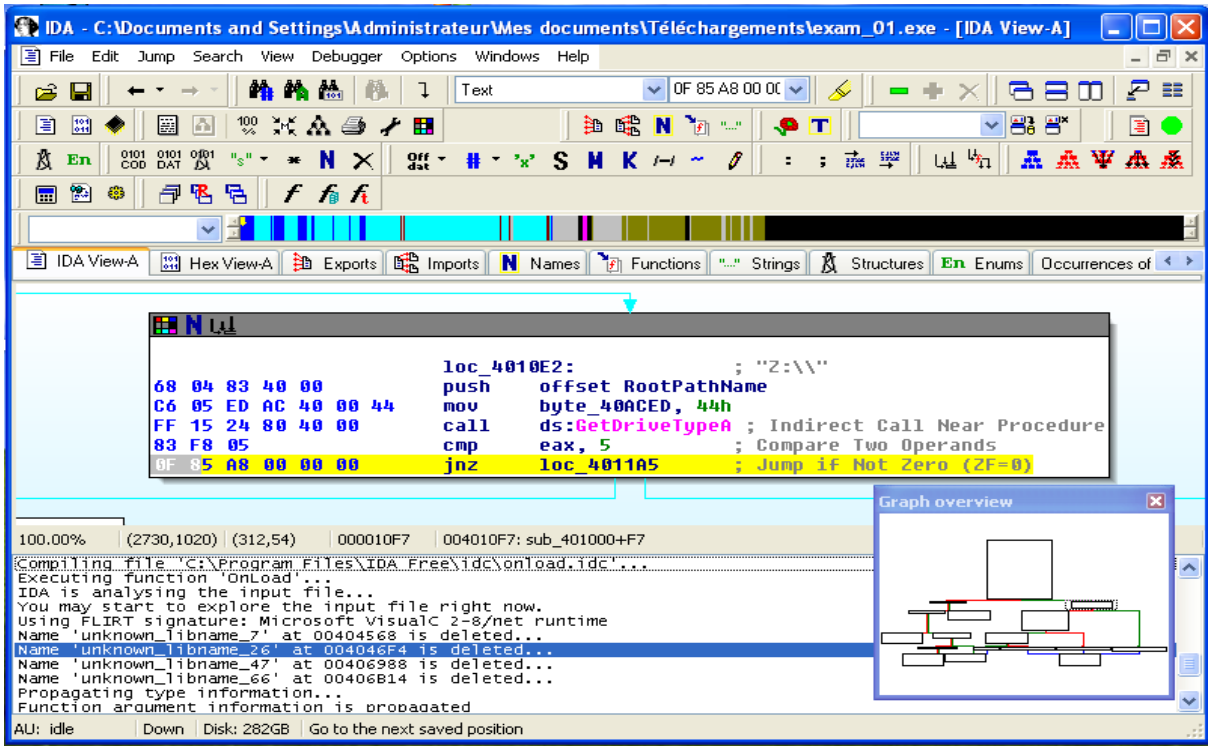
Import the file in Ida pro and you can check the jnz short loc_401226,this is the first step I found mistake,



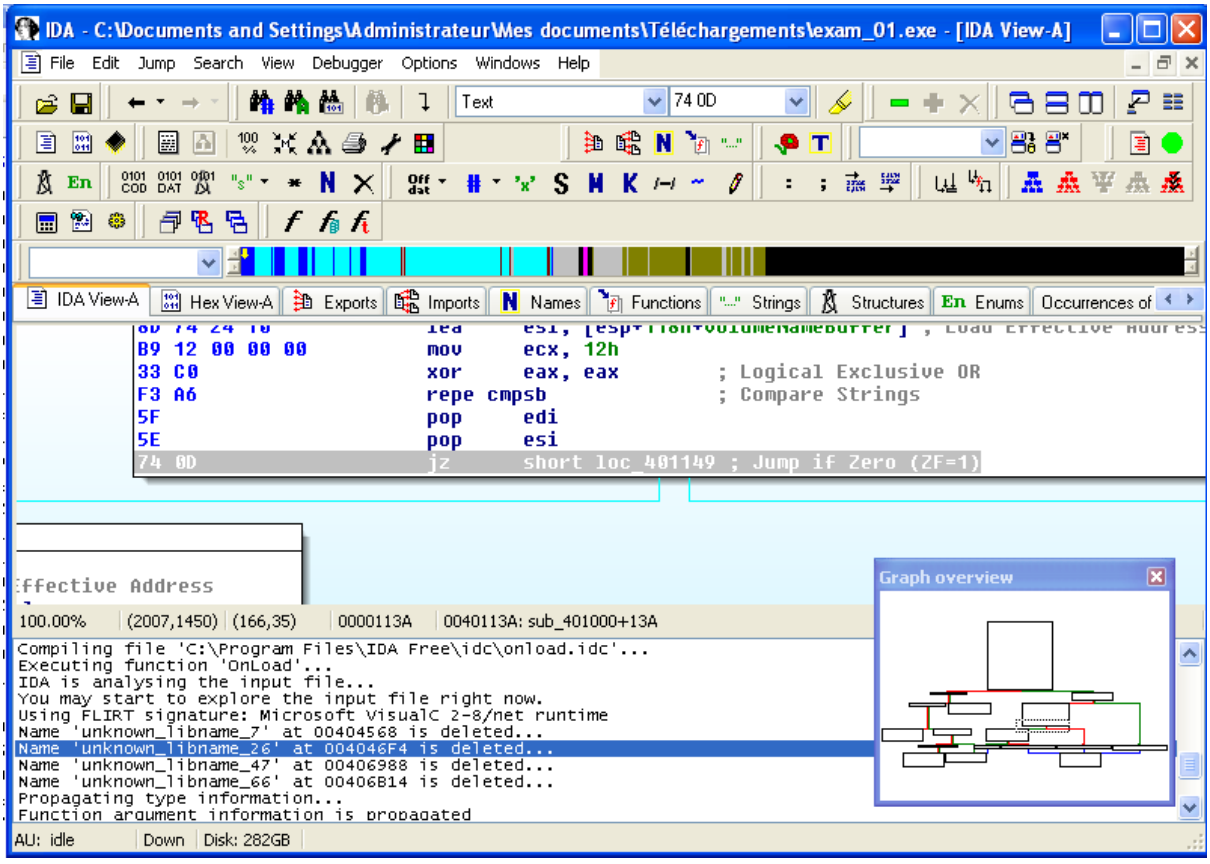
after that I found the error in hex 74 that **jz short loc_40129A**,



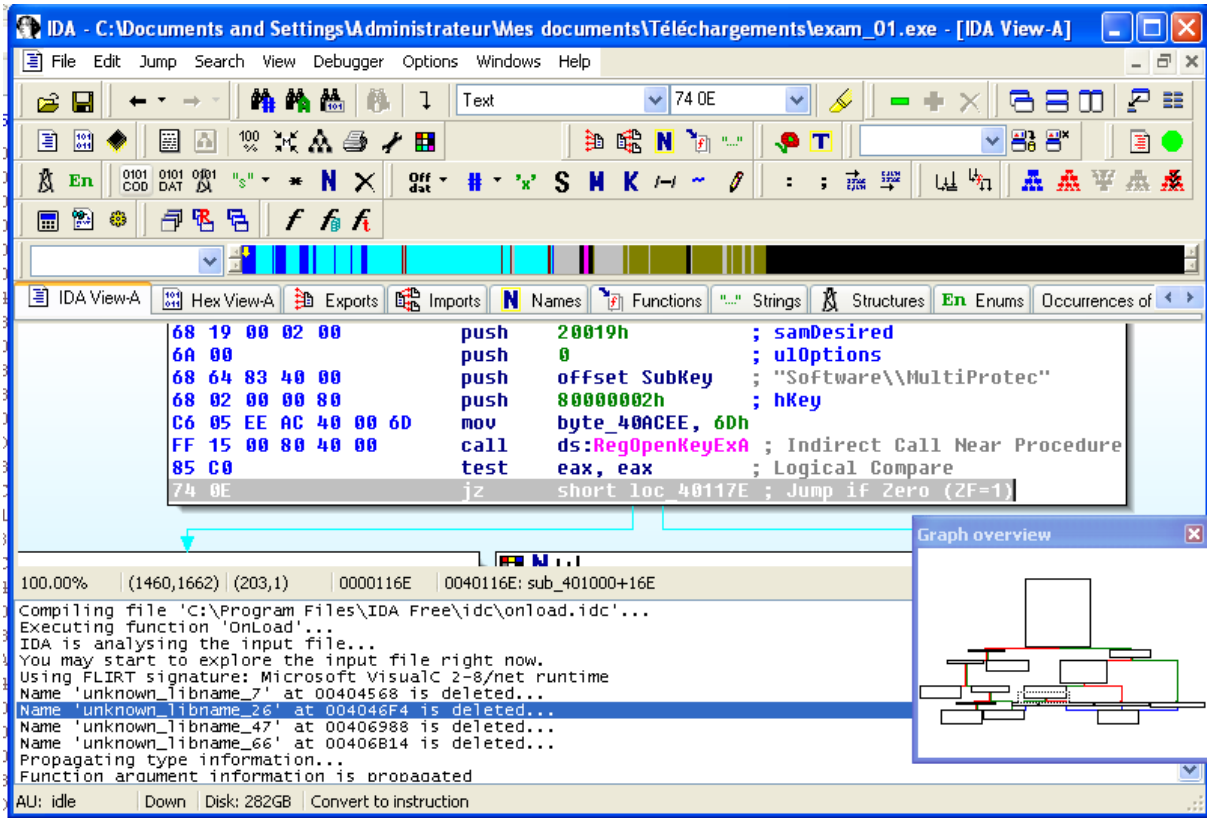
In the below screenshot we can check that the offset of **85 A8 00 00 00 jnz loc_4011A5**, we can check here it is a error,



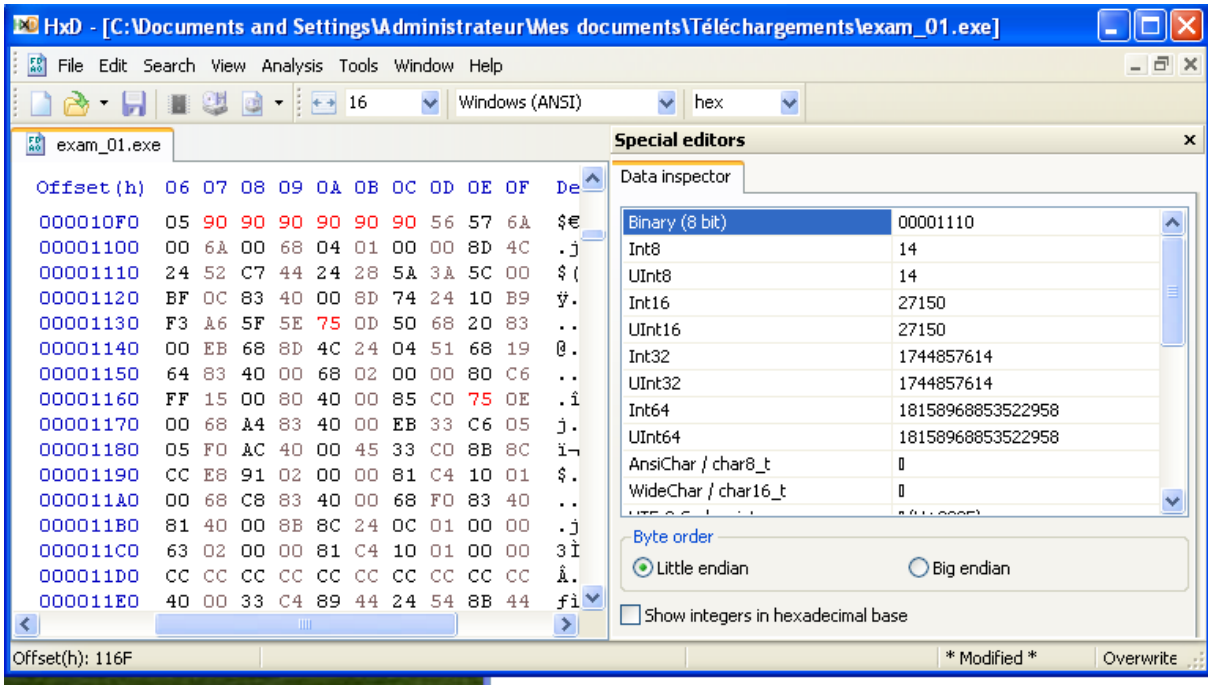
I found the error in **jz short loc_401149**, we can change it later by using hxd,



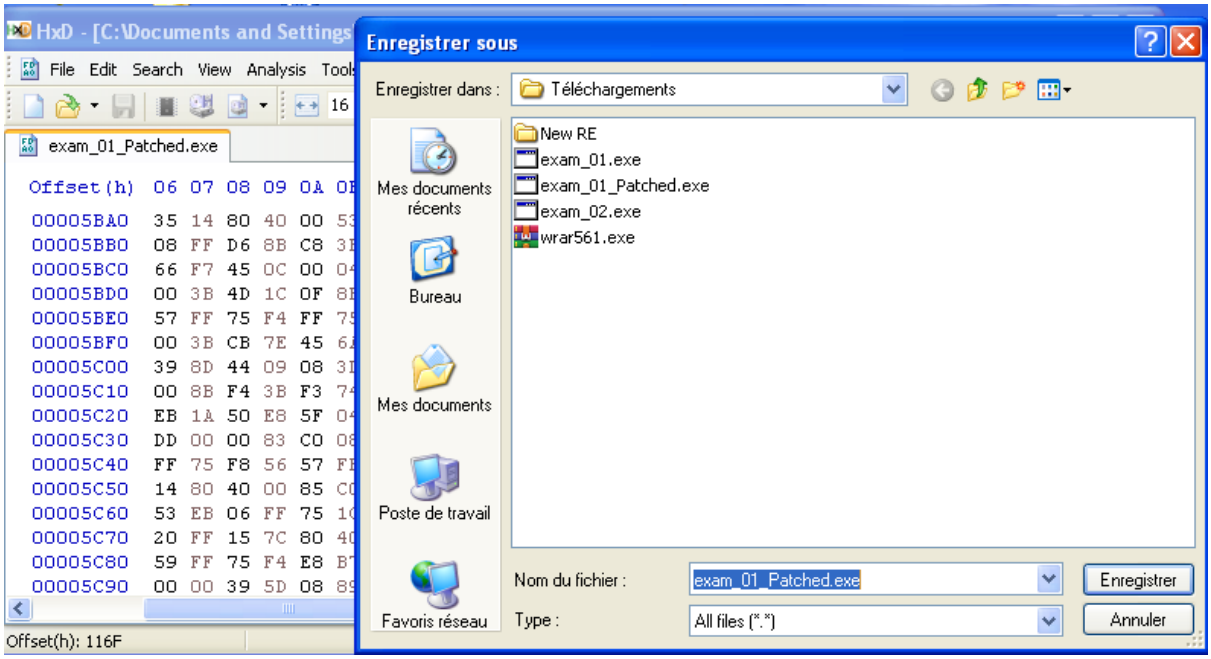
I found one more error in the below screenshot and we must change it,



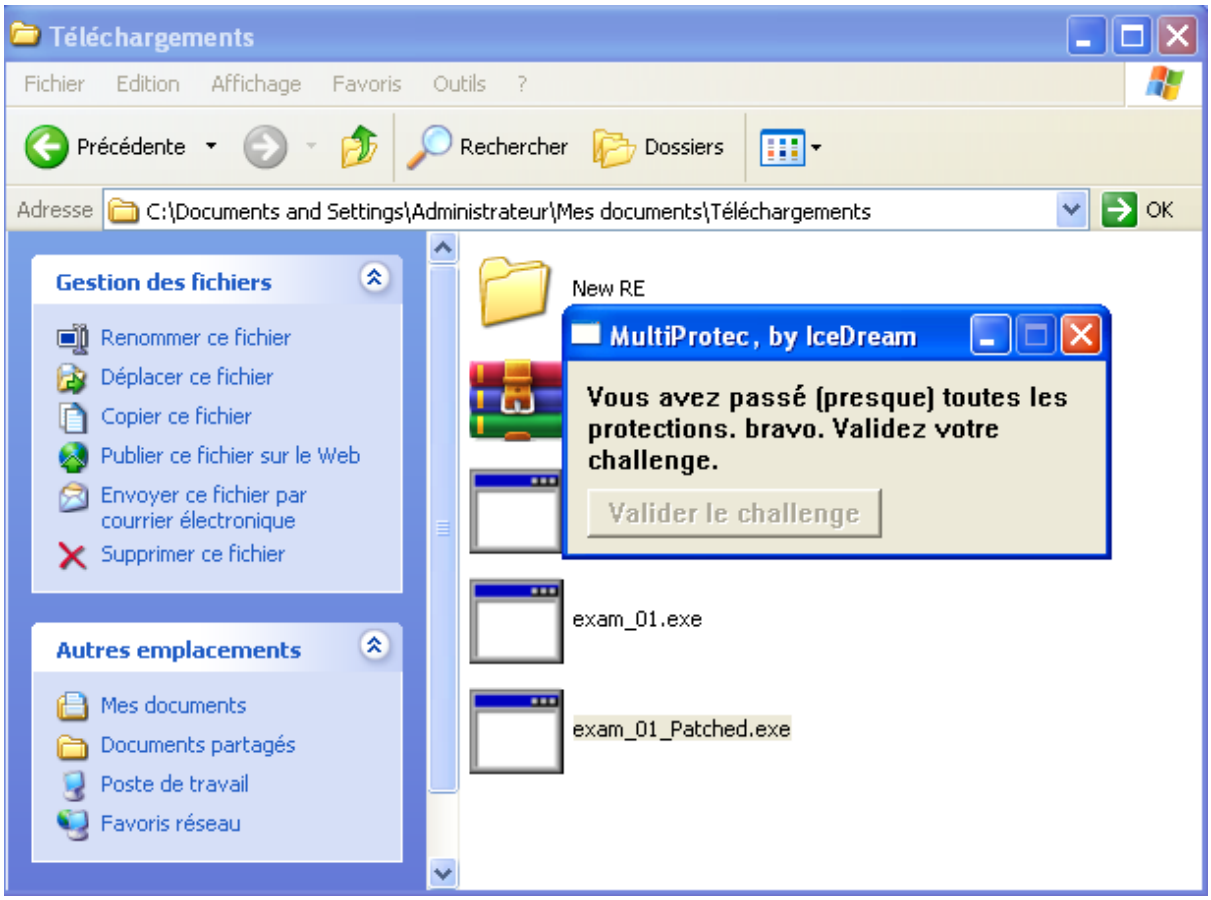
We can import the file in Hxd application, use CNTRL+G it will display search for Offset option, you can give all those offset values and edit, after changing the values it will show below screenshot,



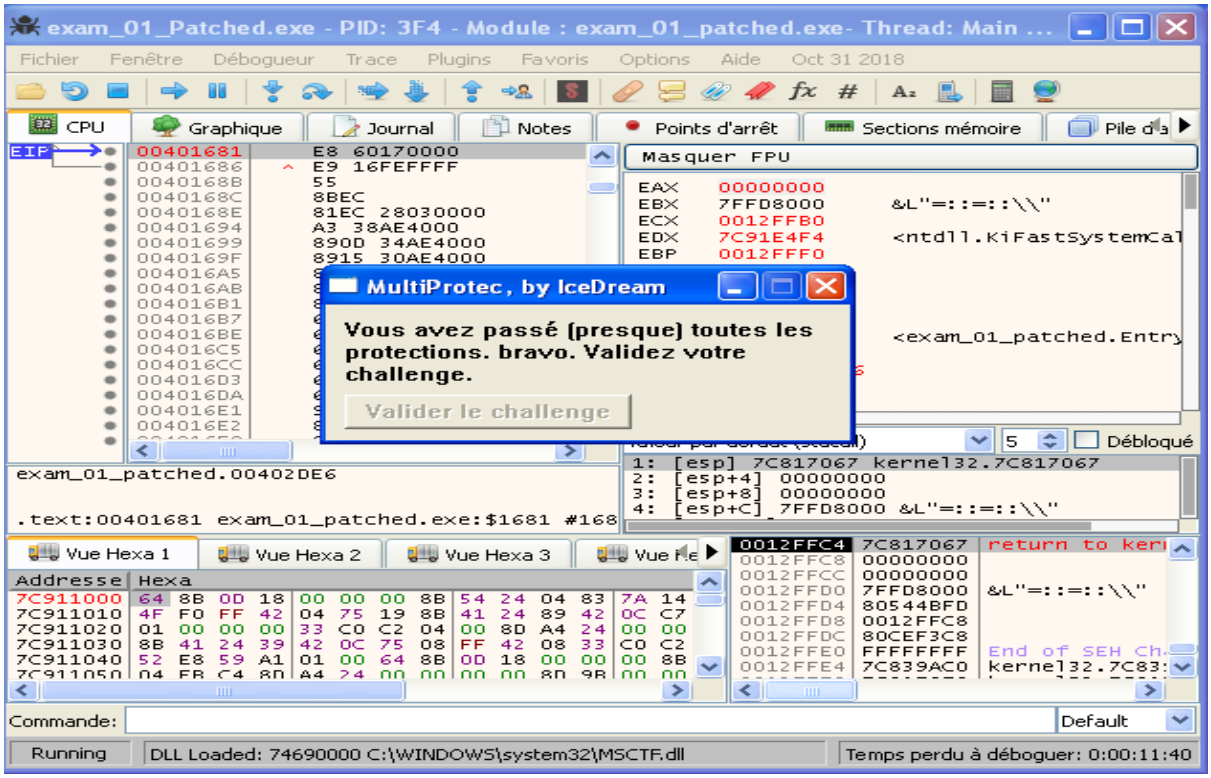
You can save the file by using save is as and give any filename (Example:exam01patched.exe),



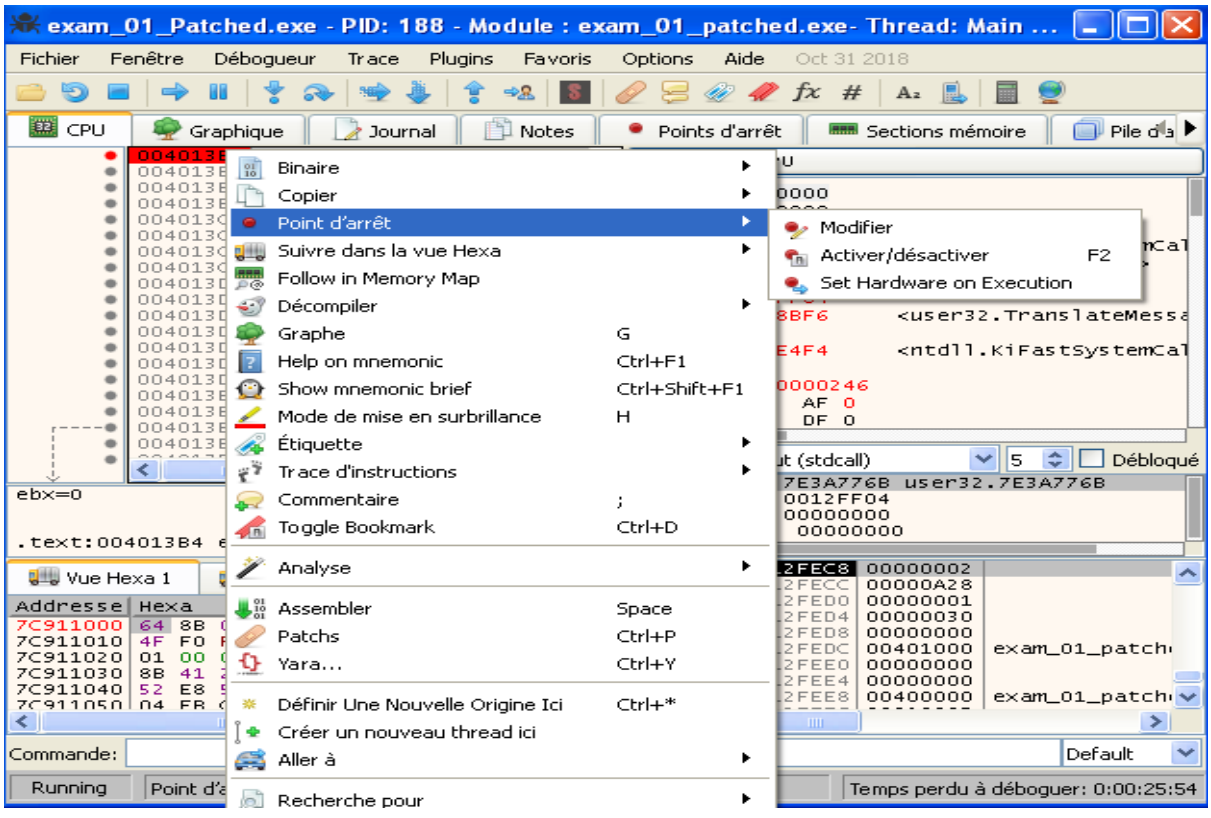
If you open the saved file, it will shown below screenshot,



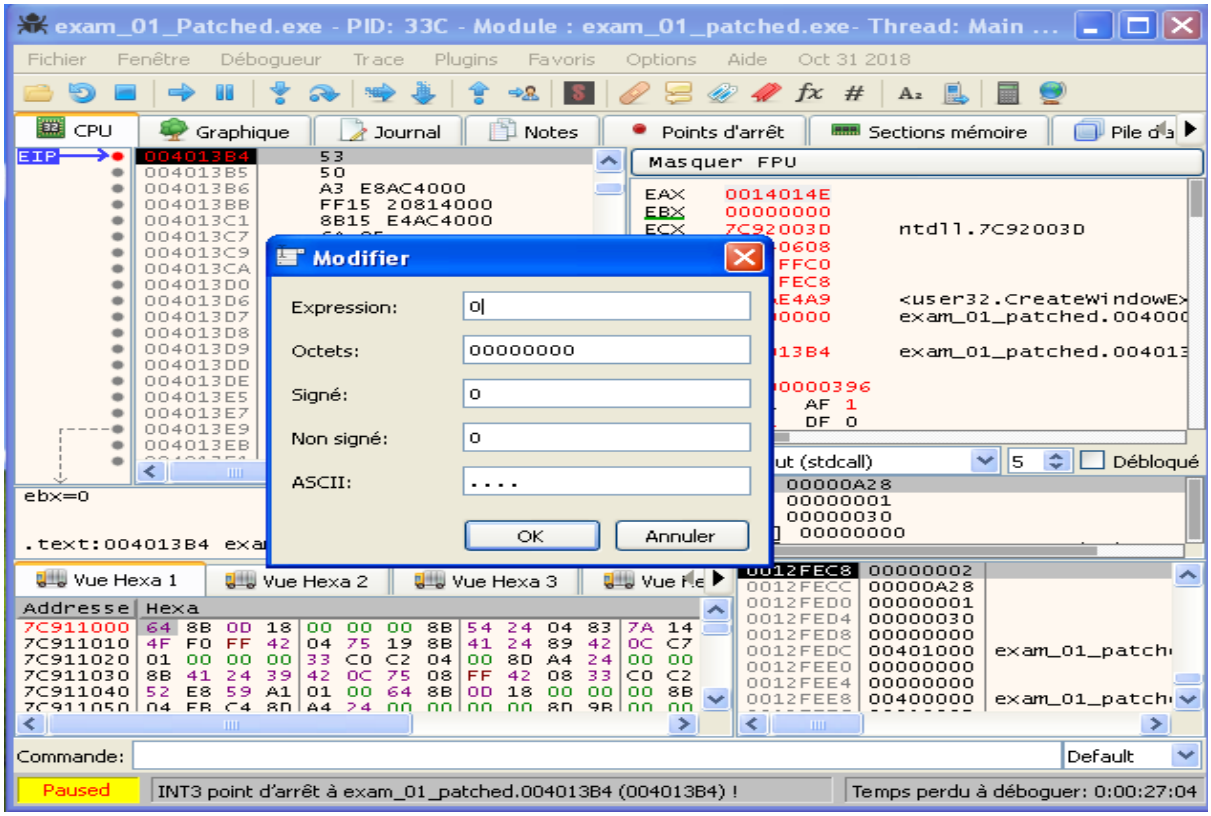
Next, you import in x32 debugger , if you click on start button, you can see the below screenshot.



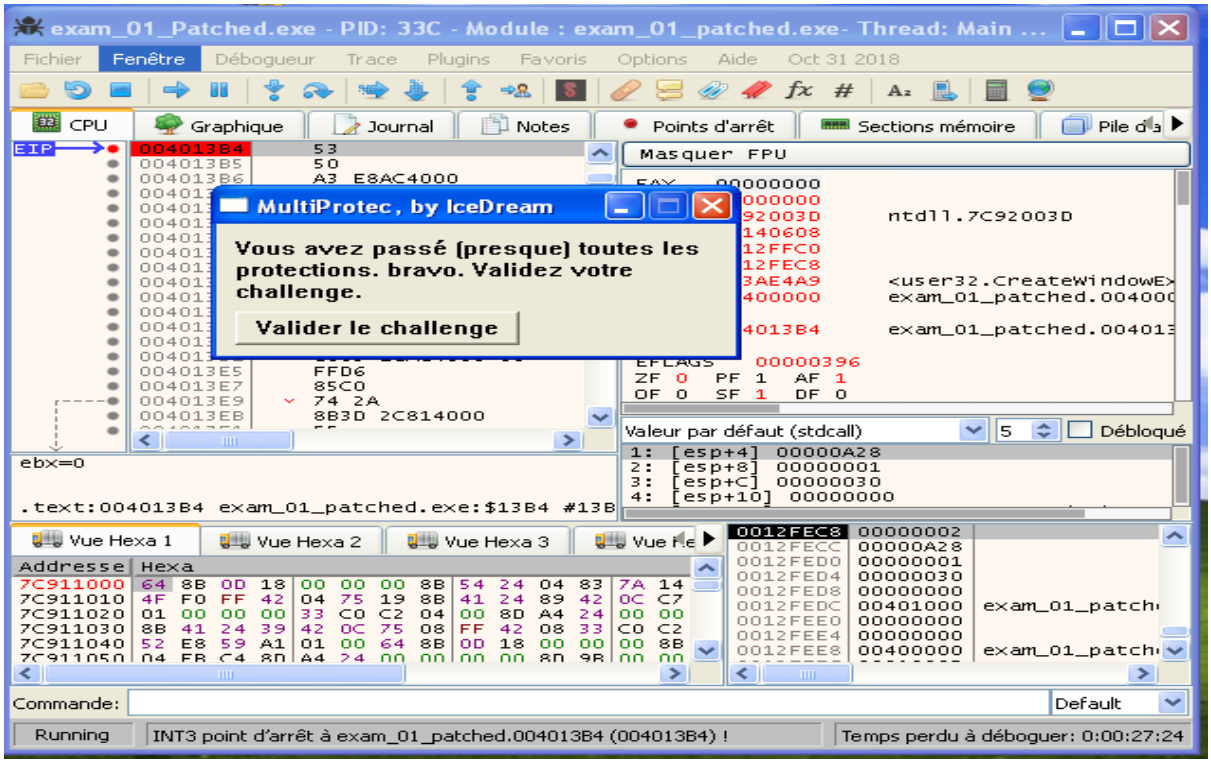
Put a breakpoint using directly F2 or using below option showing in screenshot and run the application,



After the offset 004013B4, you can found the program is mistake and you can change the eax value from 001404E to 0,



If you run a program, we can see the button showing the output like Valider challenge,



Finally if you click on valider challenge button you can see the output of your file,