GDB baby step 1

Description

Can you figure out what is in the `eax` register at the end of the `main` function? Put your answer in the picoCTF flag format: `picoCTF{n}` where `n` is the contents of the `eax` register in the decimal number base. If the answer was `0x11` your flag would be `picoCTF{17}`.

Disassemble this.

To be honest, it seems pretty straightforward.
Decompile the thing, find the main function, find eax, get the thing and convert it to decimal.

So I did exactly that.
I stumbled upon this site where I saw this :

3. Using the *gdb* Command

If we need to debug something, *gdb* is the go-to tool. Using *gdb*, we can also disassemble code:

```
$ gdb test
(gdb) disassemble main
Dump of assembler code for function main:
   0x00000000000005fa <+0>:     push   %rbp
   0x00000000000005fb <+1>:     mov    %rsp,%rbp
   0x00000000000005fe <+4>:     movl   $0x0,-0x4(%rbp)
   0x0000000000000605 <+11>:    addl   $0x14,-0x4(%rbp)
   0x0000000000000609 <+15>:    nop
   0x000000000000060a <+16>:    pop    %rbp
   0x000000000000060b <+17>:    retq
End of assembler dump.
(gdb) q
$
```

As shown above, we loaded the binary into *gdb* and **executed the *disassemble* command on the *main* function to see the assembly code**.

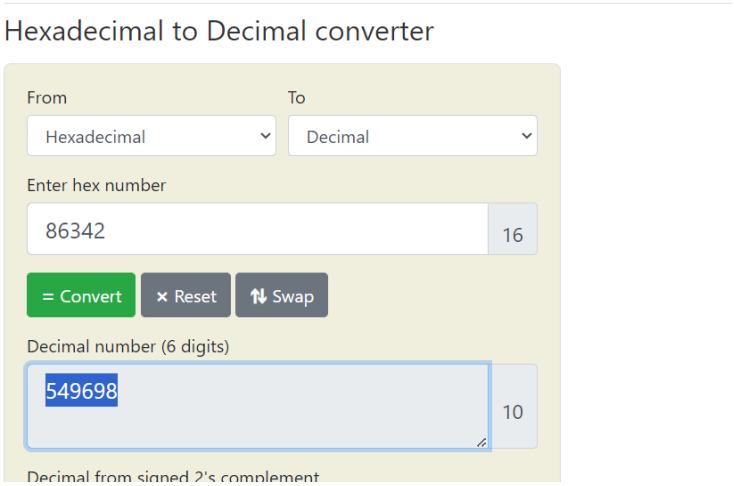
```
┌──(kali㉿kali)-[~/Desktop]
└─$ gdb debugger0_a
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from debugger0_a...
(No debugging symbols found in debugger0_a)
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000000000001129 <+0>:     endbr64
   0x000000000000112d <+4>:     push   %rbp
   0x000000000000112e <+5>:     mov    %rsp,%rbp
   0x0000000000001131 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000000000001134 <+11>:    mov    %rsi,-0x10(%rbp)
   0x0000000000001138 <+15>:    mov    $0x86342,%eax
   0x000000000000113d <+20>:    pop    %rbp
   0x000000000000113e <+21>:    ret
End of assembler dump.
(gdb)
```

*eax*ily enough (**laugh.**) :

```
0000000112d <+4>:     push   %rbp
0000000112e <+5>:     mov    %rsp,%rbp
00000001131 <+8>:     mov    %edi,-0x4(%rbp)
00000001134 <+11>:    mov    %rsi,-0x10(%rbp)
00000001138 <+15>:    mov    $0x86342,%eax
0000000113d <+20>:    pop    %rbp
0000000113e <+21>:    ret
```

Converted this to decimal



Typed in picoCTF{549698}

And yes