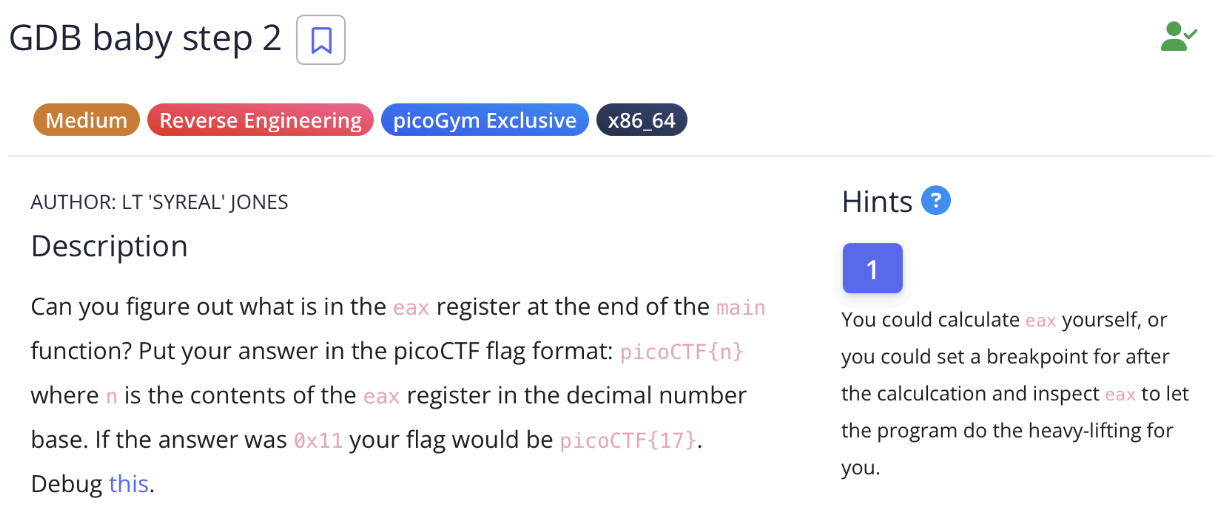

GDB baby step 2

Medium | Reverse Engineering | picoGym Exclusive | x86_64

AUTHOR: LT 'SYREAL' JONES

Description

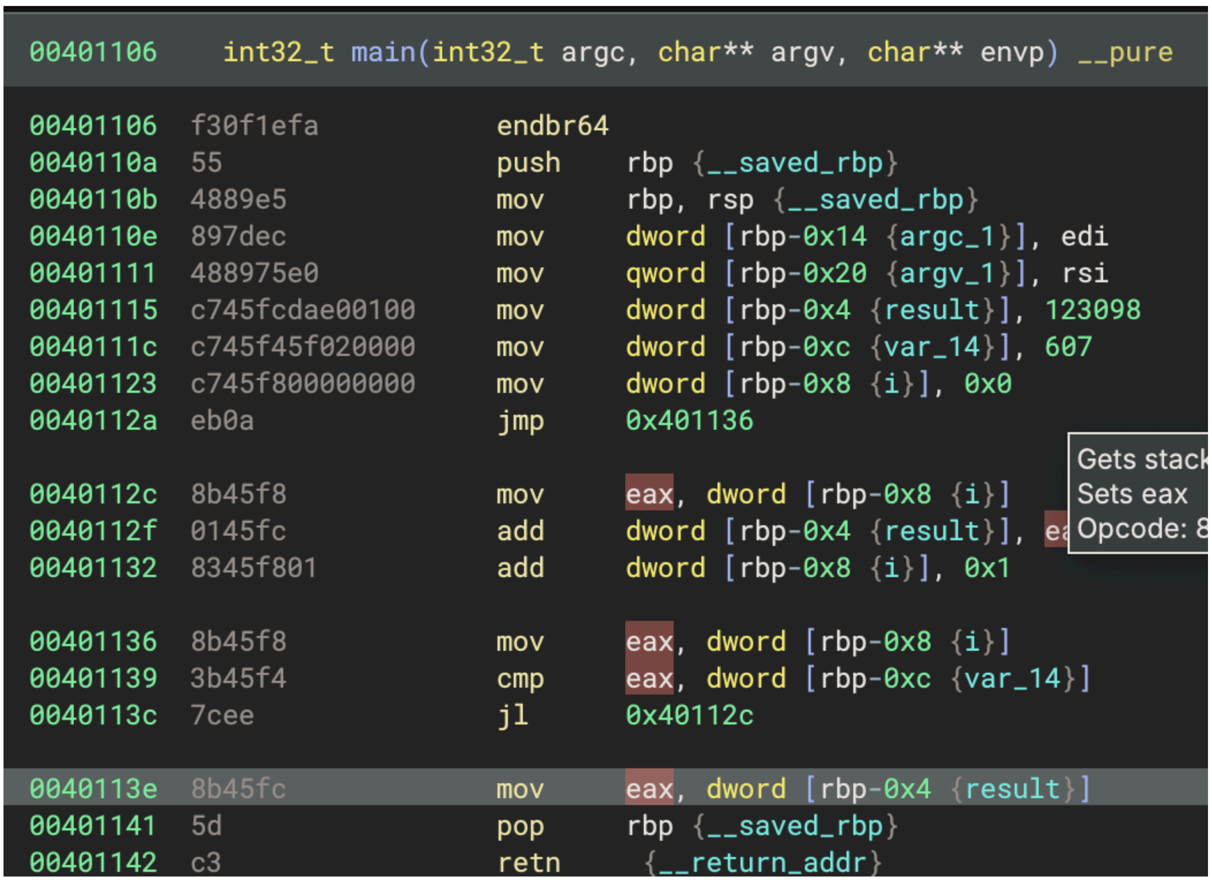
Can you figure out what is in the `eax` register at the end of the `main` function? Put your answer in the picoCTF flag format: `picoCTF{n}` where `n` is the contents of the `eax` register in the decimal number base. If the answer was `0x11` your flag would be `picoCTF{17}`.
Debug this.

Hints

1

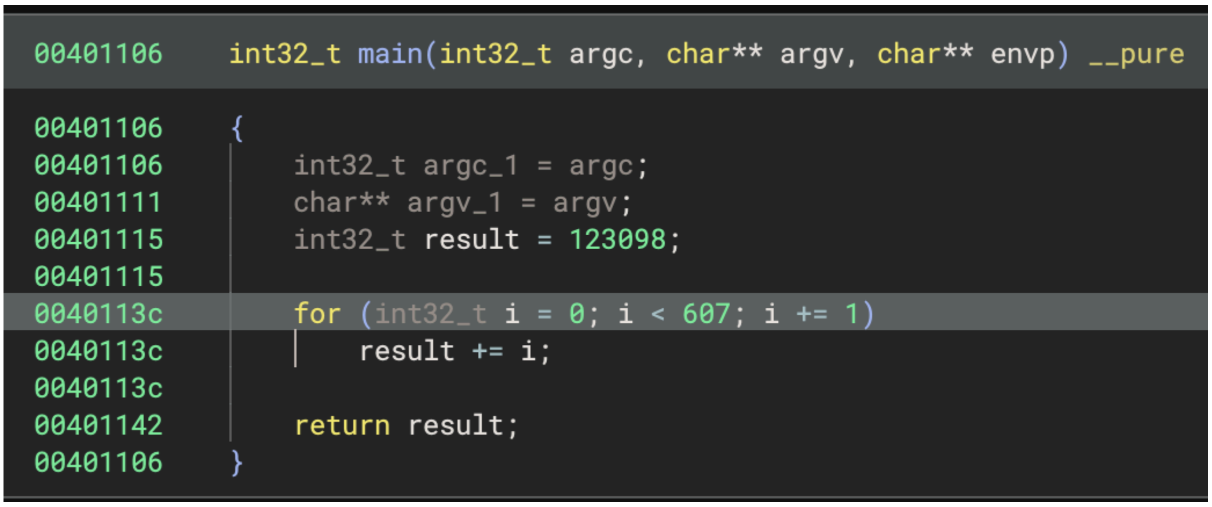
You could calculate `eax` yourself, or you could set a breakpoint for after the calculcation and inspect `eax` to let the program do the heavy-lifting for you.

Buka dengan binary ninja dan hasilnya seperti ini :



```
00401106    int32_t main(int32_t argc, char** argv, char** envp) __pure

00401106  f30f1efa        endbr64
0040110a  55              push    rbp {__saved_rbp}
0040110b  4889e5          mov     rbp, rsp {__saved_rbp}
0040110e  897dec          mov     dword [rbp-0x14 {argc_1}], edi
00401111  488975e0        mov     qword [rbp-0x20 {argv_1}], rsi
00401115  c745fcdae00100  mov     dword [rbp-0x4 {result}], 123098
0040111c  c745f45f020000  mov     dword [rbp-0xc {var_14}], 607
00401123  c745f800000000  mov     dword [rbp-0x8 {i}], 0x0
0040112a  eb0a            jmp     0x401136

0040112c  8b45f8          mov     eax, dword [rbp-0x8 {i}]      Gets stack
0040112f  0145fc          add     dword [rbp-0x4 {result}], ea   Sets eax
00401132  8345f801        add     dword [rbp-0x8 {i}], 0x1       Opcode: 8

00401136  8b45f8          mov     eax, dword [rbp-0x8 {i}]
00401139  3b45f4          cmp     eax, dword [rbp-0xc {var_14}]
0040113c  7cee            jl      0x40112c

0040113e  8b45fc          mov     eax, dword [rbp-0x4 {result}]
00401141  5d              pop     rbp {__saved_rbp}
00401142  c3              retn     {__return_addr}
```

Disini bisa kita lihat pada fungsi main bahwa variabel result disimpan pada eax, eax adalah tujuan kita pada soal ini.



```
00401106    int32_t main(int32_t argc, char** argv, char** envp) __pure

00401106    {
00401106        int32_t argc_1 = argc;
00401111        char** argv_1 = argv;
00401115        int32_t result = 123098;
00401115
0040113c        for (int32_t i = 0; i < 607; i += 1)
0040113c            result += i;
0040113c
00401142        return result;
00401106    }
```

Ini ketika kita decompile, langsung saja kita konversi ke bahasa C++

```
#include <iostream>
using namespace std;

int main(){
   int result = 123098;
   for(int i = 0; i < 607; i++){
       result += i;
   }

   cout << result;
}
```

output : 307019

Flag : picoCTF{307019}