ARMassembly 0

Description

What integer does this program print with arguments 266134863 and 1592237099? File: chall.S Flag format: picoCTF{XXXXXXXX} -> (hex, lowercase, no 0x, and 32 bits. ex. 5614267 would be picoCTF{0055aabb})

Downloaded the file chall.S, knew it was an assembly code because duh.
Searched up how to compile arm assembly code and ended up in a stack exchange where I saw these :

> I am using `as` and `gcc` to assemble and create executables of ARM assembly programs, as recommended by this tutorial, as follows:
>
> Given an assembly source file, `program.s`, I run:
>
> ```
> as -o program.o program.s
> ```
>
> Then:
>
> ```
> gcc -o program program.o
> ```

However, I ran headfirst into an error :

```
┌──(kali㉿kali)-[~/Desktop]
└─$ as -o program.o chall.s
chall.s: Assembler messages:
chall.s:1: Error: no such architecture: `armv8'
chall.s:1: Error: junk at end of line, first unrecognized character is `-'
chall.s:8: Error: expecting operand after ','; got nothing
```

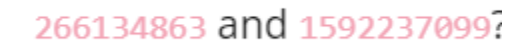
“No such architecture armv8”

I looked it up
I realised that this was an armv8 architecture code, and I was tryna compile it on my x86 device

Few minutes of googling how to cross compile armv8 on an x86 I reached this :
https://github.com/joebobmiles/ARMv8ViaLinuxCommandline

Following the instructions on there, I installed a cross compiler and an emulator to help run the arm binaries as normal executable files.

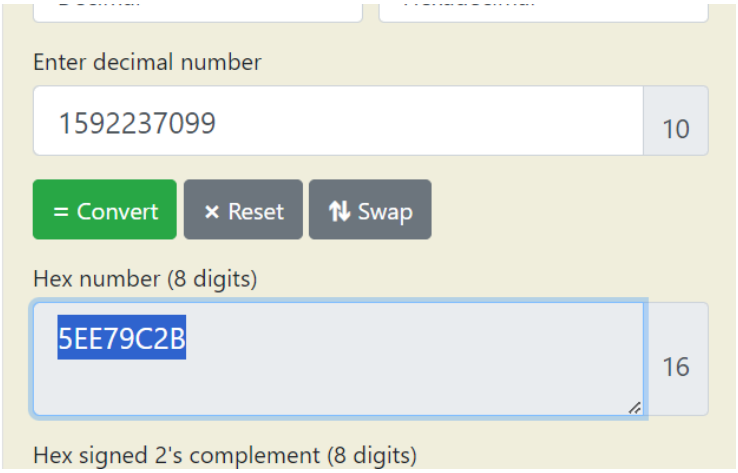
Then I ran the thing with the arguments that it had asked me to put in



Sure enough :

Yeay

I then converted this to hex



Typed in picoCTF{5EE79C2B}