

Description

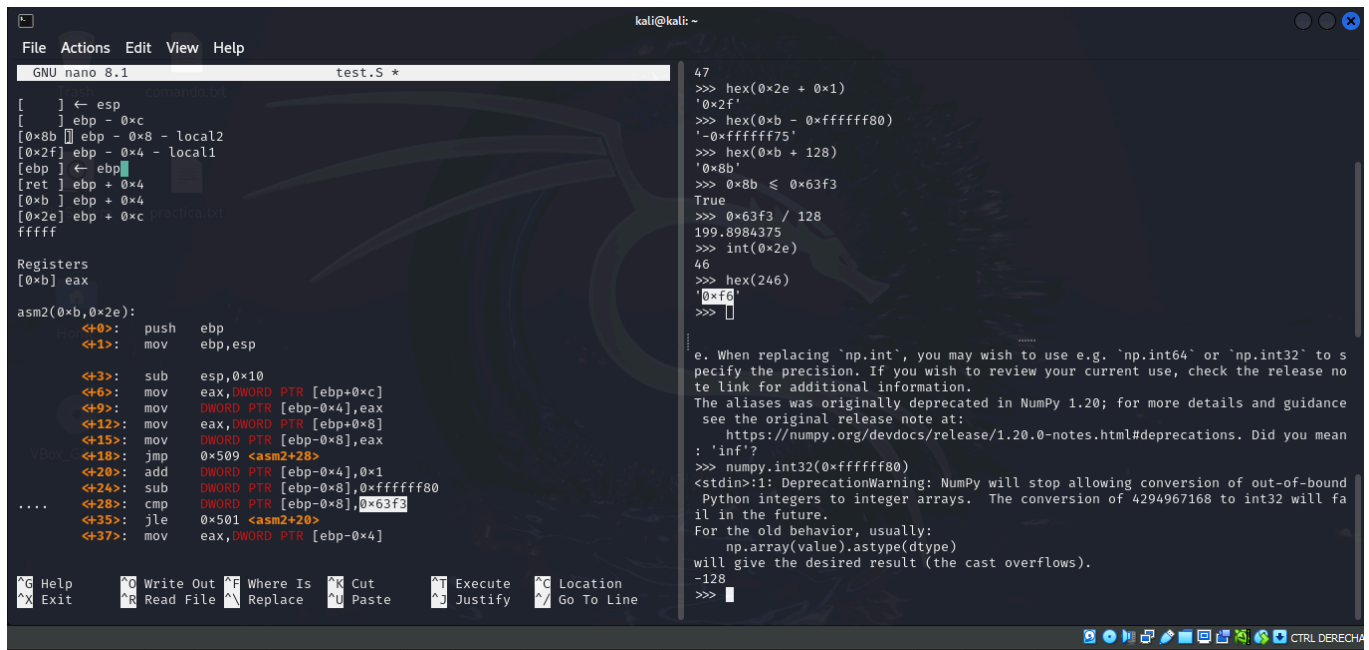
What does `asm2(0xb,0x2e)` return? Submit the flag as a hexadecimal value (starting with '0x').

NOTE: Your submission for this question will NOT be in the normal flag format. [Source](#)

Hints

- assembly [conditions](#)

Solución



```
GNU nano 8.1 test.S *
[ ] ← esp
[ ] ebp - 0xc
[0x8b] ebp - 0x8 - local2
[0x2f] ebp - 0x4 - local1
[ebp] ← ebp
[ret] ebp + 0x4
[0xb] ebp + 0x4
[0x2e] ebp + 0xc
fffff

Registers
[0xb] eax

asm2(0xb,0x2e):
<+0>: push    ebp
<+1>: mov     ebp,esp
<+3>: sub     esp,0x10
<+6>: mov     eax,DWORD PTR [ebp+0xc]
<+9>: mov     DWORD PTR [ebp+0x4],eax
<+12>: mov     eax,DWORD PTR [ebp+0x8]
<+15>: mov     DWORD PTR [ebp+0x8],eax
<+18>: jmp     0x509 <asm2+28>
<+20>: add     DWORD PTR [ebp+0x4],0x1
<+24>: sub     DWORD PTR [ebp+0x8],0xffffffff80
.... <+28>: cmp     DWORD PTR [ebp+0x8],0x63f3
<+35>: jle     0x501 <asm2+20>
<+37>: mov     eax,DWORD PTR [ebp+0x4]

e. When replacing 'np.int', you may wish to use e.g. 'np.int64' or 'np.int32' to specify the precision. If you wish to review your current use, check the release note link for additional information.
The aliases was originally deprecated in NumPy 1.20; for more details and guidance see the original release note at:
https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations. Did you mean:
'inf'?
>>> numpy.int32(0xffffffff80)
<stdin>:1: DeprecationWarning: NumPy will stop allowing conversion of out-of-bound Python integers to integer arrays. The conversion of 4294967168 to int32 will fail in the future.
For the old behavior, usually:
    np.array(value).astype(dtype)
will give the desired result (the cast overflows).
-128
>>>
```

```
(kali㉿kali)-[~/.../categorias/reversing/parte_03/asm2]
```

```
$ cat test.S
```

Stack

00000

```
[ ] ← esp
[ ] ebp - 0xc
[0x8b] ebp - 0x8 - local2
[0x2f] ebp - 0x4 - local1
[ebp] ← ebp
[ret] ebp + 0x4
[0xb] ebp + 0x4
[0x2e] ebp + 0xc
fffff
```

Registers

[0xb] eax

asm2(0xb,0x2e):

```
<+0>:  push    ebp
<+1>:  mov     ebp,esp

<+3>:  sub     esp,0x10
<+6>:  mov     eax,DWORD PTR [ebp+0xc]
<+9>:  mov     DWORD PTR [ebp-0x4],eax
<+12>: mov     eax,DWORD PTR [ebp+0x8]
<+15>: mov     DWORD PTR [ebp-0x8],eax
<+18>: jmp     0x509 <asm2+28>
<+20>: add     DWORD PTR [ebp-0x4],0x1
<+24>: sub     DWORD PTR [ebp-0x8],0xffffffff80
....  <+28>: cmp     DWORD PTR [ebp-0x8],0x63f3
<+35>: jle     0x501 <asm2+20>
<+37>: mov     eax,DWORD PTR [ebp-0x4]

<+40>: leave
<+41>: ret
```

└─(kali⊗kali)-[~]

└─\$ python3

Python 3.11.9 (main, Apr 10 2024, 13:16:36) [GCC 13.2.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> int(0x10)
```

```
16
```

```
>>> 0xb <= 0x63f3
```

```
True
```

```
>>> 0x2e + 0x1
```

```
47
```

```
>>> hex(0x2e + 0x1)
```

```
'0x2f'
```

```
>>> hex(0xb - 0xffffffff80)
```

```
'-0xffffffff75'
```

```
>>> hex(0xb + 128)
```

```
'0x8b'
```

```
>>> 0x8b <= 0x63f3
```

```
True
```

```
>>> 0x63f3 / 128
```

```
199.8984375
```

```
>>> int(0x2e)
```

```
46
```

```
>>> hex(246)
```

```
'0xf6'
```

```
>>>
```

```
└─(kali㉿kali)-[~]
```

```
└─$ python3
```

```
Python 3.11.9 (main, Apr 10 2024, 13:16:36) [GCC 13.2.0] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import numpy
```

```
>>> numpy.int32(0xffffffff80)
```

```
<stdin>:1: DeprecationWarning: NumPy will stop allowing conversion of out-of-bound Python integers to integer arrays. The conversion of 4294967168 to int32 will fail in the future.
```

```
For the old behavior, usually:
```

```
    np.array(value).astype(dtype)
```

```
will give the desired result (the cast overflows).
```

```
-128
```

```
>>>
```

Bandera

```
flag: 0xf6
```

Notas Adicionales

Referencias

-