About to call printf()

```
0x40055f <main+8>          lea    rdi, [rip+0xd2]        # 0x400638
0x400566 <main+15>         mov    eax, 0x0
0x40056b <main+20>         call   0x400450 <printf@plt>
```

In this case only 1 argument is supplied

```
printf@plt (
   $rdi = 0x0000000000400638 → "Type your favorite keyboard character:",
   $rsi = 0x00007fffffffe508 → 0x00007fffffffe746 → "/home/tao/cprog/chap9/ex1"
)
```

```
      2
      3  int main()
      4  {
      5      char key;
      6
 →    7      printf("Type your favorite keyboard character: ");
```

About to call scanf()

```
0x400570 <main+25>         lea    rax, [rbp-0x1]
0x400574 <main+29>         mov    rsi, rax
0x400577 <main+32>         lea    rdi, [rip+0xe2]        # 0x400660
0x40057e <main+39>         mov    eax, 0x0
0x400583 <main+44>         call   0x400460 <__isoc99_scanf@plt>
```

Rdi holds the first argument -> format specifier

Rsi holds the second argument -> the buffer in the stack to store the said data

```
__isoc99_scanf@plt (
   $rdi = 0x0000000000400660 → 0x0000000000006325 ("%c"?),
   $rsi = 0x00007fffffffe41f → 0x000000004005b000,
   $rdx = 0x0000000000000000
)
```

Here we see the 0x41 which translates to **A** right before **rbp**

```
gef➤  x/gx $rbp
0x7fffffffe420: 0x00000000004005b0
gef➤
```

```
gef➤  x/4gx $rbp-16
0x7fffffffe410: 0x00007fffffffe500      0x4100000000000000
0x7fffffffe420: 0x00000000004005b0      0x00007ffff7a05b97
gef➤
```