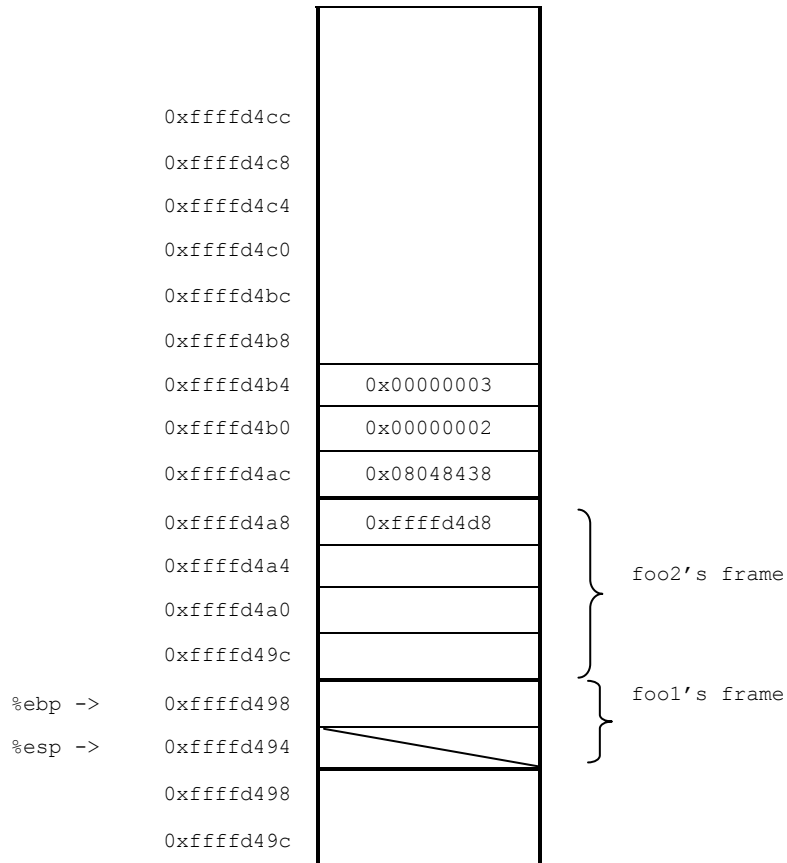


Stack Frame Structure

Fill in the missing part of the stack when the given program just entered `foo1()`, by examining the attached C source code and assembly code.

Assume the two inputs from the user are 2 and 3.



[C code]

```
#include <stdio.h>

int fool(int *xp, int *yp) {

    int x = *xp;
    int y = *yp;

    *xp = y;
    *yp = x;

    return x + y;
}

int foo2(int x, int y) {

    int sum = fool(&x, &y);
    int dif = x - y;

    return dif * sum;
}

int main(void)
{
    int x, y;
    int val;

    scanf("%d %d", &x, &y);

    val = foo2(x, y);

    printf("val=%d\n", val);

    return 0;
}
```

[ASM code]

```
080483b0 <fool>:
80483b0: 55                push   %ebp
80483b1: 89 e5            mov    %esp,%ebp
80483b3: 53              push   %ebx
80483b4: 8b 4d 08        mov    0x8(%ebp),%ecx
80483b7: 8b 5d 0c        mov    0xc(%ebp),%ebx
80483ba: 8b 01          mov    (%ecx),%eax
80483bc: 8b 13          mov    (%ebx),%edx
80483be: 89 11          mov    %edx,(%ecx)
80483c0: 89 03          mov    %eax,(%ebx)
80483c2: 01 d0          add    %edx,%eax
80483c4: 5b              pop    %ebx
80483c5: c9              leave
80483c6: c3              ret
80483c7: 89 f6          mov    %esi,%esi
80483c9: 8d bc 27 00 00 00 00 lea    0x0(%edi),%edi
```

```

080483d0 <foo2>:
80483d0: 55                push   %ebp
80483d1: 89 e5            mov    %esp,%ebp
80483d3: 83 ec 08        sub    $0x8,%esp
80483d6: 8d 45 0c        lea   0xc(%ebp),%eax
80483d9: 89 44 24 04     mov    %eax,0x4(%esp)
80483dd: 8d 45 08        lea   0x8(%ebp),%eax
80483e0: 89 04 24        mov    %eax,(%esp)
80483e3: e8 c8 ff ff ff  call   80483b0 <foo1>
80483e8: 89 c2            mov    %eax,%edx
80483ea: 8b 45 08        mov    0x8(%ebp),%eax
80483ed: 2b 45 0c        sub    0xc(%ebp),%eax
80483f0: c9              leave
80483f1: 0f af c2       imul  %edx,%eax
80483f4: c3              ret
80483f5: 8d 74 26 00     lea   0x0(%esi),%esi
80483f9: 8d bc 27 00 00 00 00  lea   0x0(%edi),%edi

08048400 <main>:
8048400: 55                push   %ebp
8048401: 89 e5            mov    %esp,%ebp
8048403: 83 ec 18        sub    $0x18,%esp
8048406: 8d 45 fc        lea   0xffffffffc(%ebp),%eax
8048409: 83 e4 f0        and   $0xffffffff0,%esp
804840c: 83 ec 10        sub    $0x10,%esp
804840f: 89 44 24 08     mov    %eax,0x8(%esp)
8048413: 8d 45 f8        lea   0xffffffff8(%ebp),%eax
8048416: c7 04 24 3c 85 04 08  movl  $0x804853c,(%esp)
804841d: 89 44 24 04     mov    %eax,0x4(%esp)
8048421: e8 9e fe ff ff  call   80482c4 <scanf@plt>
8048426: 8b 45 fc        mov    0xffffffffc(%ebp),%eax
8048429: 89 44 24 04     mov    %eax,0x4(%esp)
804842d: 8b 45 f8        mov    0xffffffff8(%ebp),%eax
8048430: 89 04 24        mov    %eax,(%esp)
8048433: e8 98 ff ff ff  call   80483d0 <foo2>
8048438: c7 04 24 42 85 04 08  movl  $0x8048542,(%esp)
804843f: 89 44 24 04     mov    %eax,0x4(%esp)
8048443: e8 9c fe ff ff  call   80482e4 <printf@plt>
8048448: c9              leave
8048449: 31 c0            xor    %eax,%eax
804844b: c3              ret

```