Declaration of integer variables and assigning them values

```
→   0x4004ef <main+8>          mov    DWORD PTR [rbp-0x4], 0x1
    0x4004f6 <main+15>         mov    DWORD PTR [rbp-0x8], 0x2
```

Moving of integer variable values from memory to register and passing them as arguments into functions

Rdi = 1st argument

Rsi = 2nd argument

```
0x4004fd <main+22>         mov    edx, DWORD PTR [rbp-0x8]
0x400500 <main+25>         mov    eax, DWORD PTR [rbp-0x4]
0x400503 <main+28>         mov    esi, edx
0x400505 <main+30>         mov    edi, eax
```

Creating a new stack frame

Rbp - 0x4 = 1st argument

Rbp - 0x8 = 2nd argument

```
0x400533 <add+1>           mov    rbp, rsp
0x400536 <add+4>           mov    DWORD PTR [rbp-0x4], edi
0x400539 <add+7>           mov    DWORD PTR [rbp-0x8], esi
```

Edx = 1

Eax = 2

Eax = 1 + 2

Return value will be stored in eax

```
→   0x40053c <add+10>          mov    edx, DWORD PTR [rbp-0x4]
    0x40053f <add+13>          mov    eax, DWORD PTR [rbp-0x8]
    0x400542 <add+16>          add    eax, edx
```

0x8 + 0x4 = 0xC

```
→   0x40050c <main+37>          mov    DWORD PTR [rbp-0xc], eax
```

End result

```
printf@plt (
   $rdi = 0x00000000004005d4 → "%d + %d = %d",
   $rsi = 0x0000000000000001,
   $rdx = 0x0000000000000002,
   $rcx = 0x0000000000000003
)
```