

Problem 1:

Registers after the number variable is moved into eax with the value of -105.

```
Registers
EAX = 008FF89C EBX = 00775000 ECX = 00281005 EDX = 00281005 ESI = 00281005 EDI = 00281005 EIP = 00281010
ESP = 008FF848 EBP = 008FF854 EFL = 00000246

0x00284000 = FFFFFFF97
```

Registers after the number2 variable is moved into eax with the value of 105.

```
Registers
EAX = FFFFFFF97 EBX = 00775000 ECX = 00281005 EDX = 00281005 ESI = 00281005 EDI = 00281005 EIP = 00281015
ESP = 008FF848 EBP = 008FF854 EFL = 00000246

0x00284004 = 00000069
```

Registers after the sum of eax is taken and put into memory.

```
Registers
EAX = 00000000 EBX = 00775000 ECX = 00281005 EDX = 00281005 ESI = 00281005 EDI = 00281005 EIP = 00281018
ESP = 008FF848 EBP = 008FF854 EFL = 00000257

0x00284008 = 00000000
```

Registers during the exit code of 0.

```
Registers
EAX = 00000000 EBX = 00775000 ECX = 00281005 EDX = 00281005 ESI = 00281005 EDI = 00281005 EIP = 00281020
ESP = 008FF848 EBP = 008FF854 EFL = 00000257
```

Problem 2:

Registers and Memory after taking the first input (10), this is stored in memory.

```
Registers
EAX = 0000000A EBX = 0030C000 ECX = 00000000 EDX = 004F9A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411D3D
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000206

0x0041A000 = 00000000
```

Registers and Memory after taking the second input (15), this is stored in memory.

```
Registers
EAX = 0000000F EBX = 002F1000 ECX = 00000000 EDX = 00579A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411D6F
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000206

0x0041A004 = 00000000
```

Registers and Memory after taking the third input (20), this is also stored into memory.

```
Registers
EAX = 00000014 EBX = 002F1000 ECX = 00000000 EDX = 00579A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411DA1
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000206

0x0041A008 = 00000000
```

First input is moved into eax

```
Registers
EAX = 00000014 EBX = 002F1000 ECX = 00000000 EDX = 00579A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411DA6
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000206

0x0041A008 = 0000000A
```

The second number is added to the first number

```
Registers
EAX = 0000000A EBX = 002F1000 ECX = 00000000 EDX = 00579A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411DB
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000206

0x0041A004 = 0000000F
```

The third number is added to the previous two

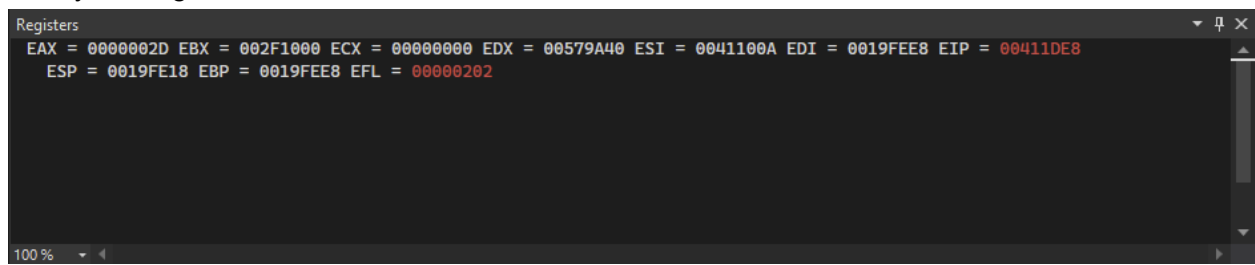
```
Registers
EAX = 00000019 EBX = 002F1000 ECX = 00000000 EDX = 00579A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411DB1
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000212

0x0041A008 = 00000014
```

All of the numbers are summed together to equal 2D (45 in decimal)

```
Registers
EAX = 0000002D EBX = 002F1000 ECX = 00000000 EDX = 00579A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411DB7
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000206
```

Finally the registers after it exits with a return code of 0.



A screenshot of a debugger's 'Registers' window. The window has a title bar with 'Registers' and standard window controls. The background is dark gray. The register values are displayed in a light gray font. The EIP register value '00411DE8' and the EFL register value '00000202' are highlighted in red. A vertical scrollbar is on the right side. The bottom status bar shows '100 %' and a zoom icon.

```
Registers
EAX = 0000002D EBX = 002F1000 ECX = 00000000 EDX = 00579A40 ESI = 0041100A EDI = 0019FEE8 EIP = 00411DE8
ESP = 0019FE18 EBP = 0019FEE8 EFL = 00000202
100 %
```