

Exercises:

1. Fill in the blank (”English Equivalent”) on the table above.

|  |  |
| --- | --- |
| **x86 instructions** | **English equivalent** |
| movq $351, %rax | Move the number 351 into 8-byte (quad) register “rax” |
| addq %rdi, %rsi | Add the 64-bit value of %rdi to %rsi |
| movq (%rdi), %r8 | Move the 64-bit data at the address stored in %rdi to %r8 |
| leaq (%rax,%rax,8), %rax | Compute 9 \* %rax, and store the 64-bit result in %rax |

2. Symbolically, what does the following code return?

**movl** (%rdi), %eax # %rdi -> x

**r = \*x**

**leal** (%eax,%eax,2), %eax # %rax -> r

**r = (\*x) \* 3**

**addl %eax, %eax**

**#r = (\*x)\*3 + (\*x)\*3**

**andl** %esi, %eax # %rsi -> y

**#r = ((\*x)\*6) & y**

**subl %esi, %eax**

**#r = (((\*x)\*6) & y) - y**

**ret**

**(((\*x) \* 6) & y) - y**

3. Convert the following C function into x86-64 assembly code. You are not being judged on the efficiency of your code – just the correctness. Use this Web Site to accomplish the task: <https://godbolt.org/>

**long happy(long \*x, long y, long z) {**

if (y > z)

return z + y;

else

return \*x;

}

**happy:**

**cmpq %rdx, %rsi jle .else**

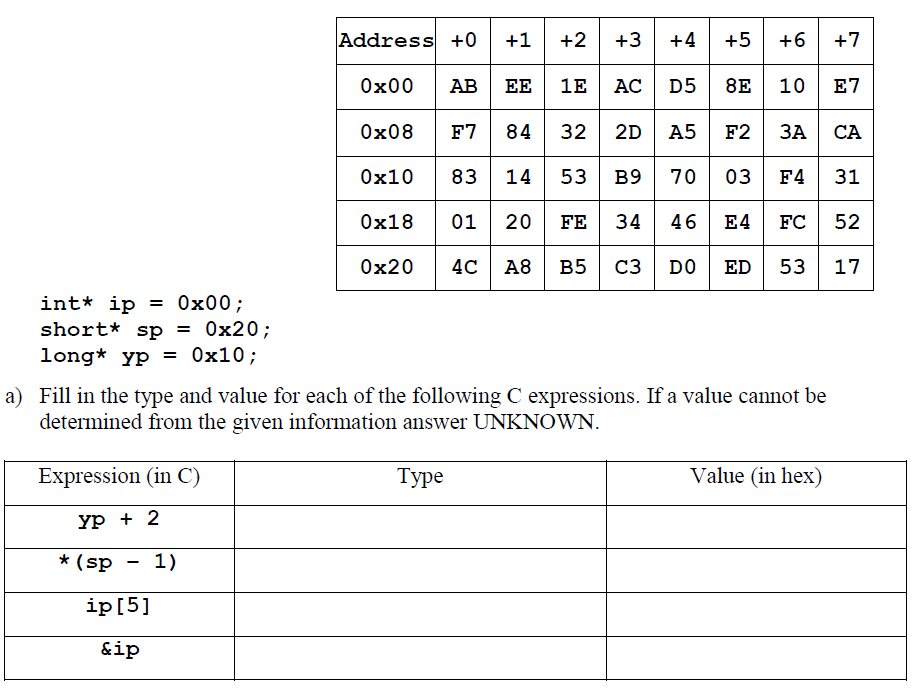
**leaq (%rdx, %rsi), %rax ret**

**.else:**

**movq (%rdi), %rax ret**

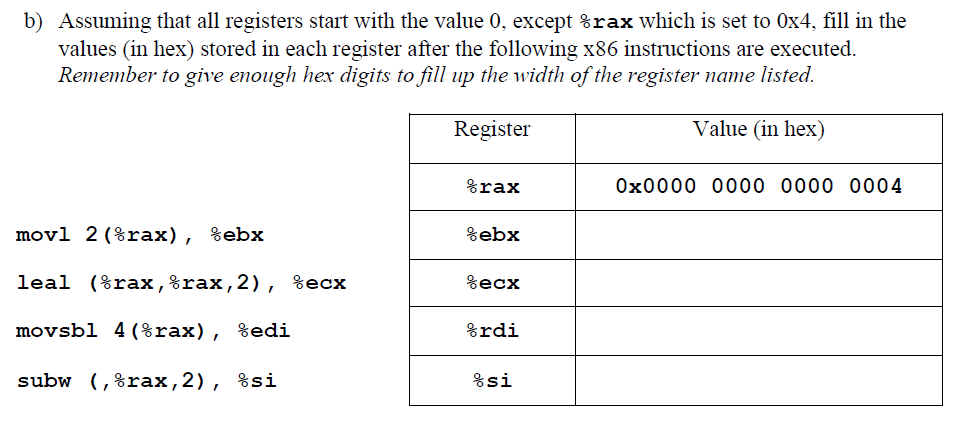
4. Assuming a 64-bit x86-64 machine (little endian), you are given the following variables and

initial state of memory (values in hex) shown below:

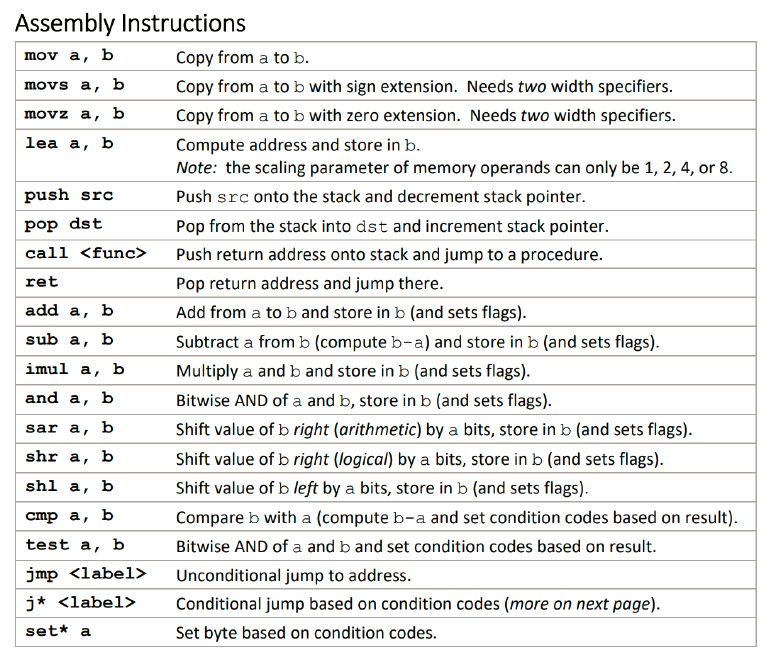


|  |  |  |
| --- | --- | --- |
| **Expression (in C)** | **Type** | **Value (in hex)** |
|  |  |  |
| **yp + 2** | **long\*** | **0x20** |
|  |  |  |
| **\*(sp – 1)** | **short** | **0x52FC** |
| **ip[5]** | **int** | **0x31F40370** |
|  |  |  |
| **&ip** | **int\*\*** | **UNKNOWN** |
|  |  |  |

Notes: ip, sp, yp denote the corresponding memory locations. It is pointers (covered further next week).



|  |  |
| --- | --- |
| Register | Value (in hex) |
| %rax | **0x0000 0000 0000 0004** |
| %ebx | **0x84F7 e710** |
| %ecx | **0x0000 000c** |
| %rdi | **0x0000 0000 ffff fff7** |
| %si | **0x7B09** |



Notes:

* To insert diagram in MS Word, use its “Insert” -> “Shape” feature.
* To add text onto the image, please see the tutorial video found on Lab1 assignment.

Lab Results Submission:

Please submit your results online. If that fails, email me.  The subject of the email should be:  [Your StudentID, Assignment Subject].  e.g. [12345678, Lab\_x86 Programming].