Solutions to Practice Problems

Practice Problem 3.1
For each of the for loops shown below, state how many times the loop will iterate.

(a) \( \text{for} \ (i = 1 \ ; \ i \leq 100 \ ; \ i = i + 1) \)
(b) \( \text{for} \ (i = 3 \ ; \ i > 1 \ ; \ i = i - 1) \)
(c) \( \text{for} \ (i = 7 \ ; \ i \leq 21 \ ; \ i = i + 7) \)

Solution:
(a) 100  (b) 2  (c) 3

Practice Problem 3.2
Examine the following C program and describe the expected output.

```c
#include<stdio.h>
int main( )
{
    int count;
    for( count = 1 ; count <= 2 ; count = count + 1 )
    {
        if( count > 1 )
            printf( "Cyber\n" );
        else
            printf( "Fun\n" );
    }
}
```

Solution:

midshipman@EC310:/work $ ./a.out
Fun
Cyber
midshipman@EC310:/work $ [output]

Practice Problem 3.3
How many bits are in an address?

Solution: Look at the first address: 08048374. This is 8 hex digits. Each hex digit is 4 bits. So 8 hex digits are 32 bits. Addresses are 32 bits.

Practice Problem 3.4
How many bits are represented by the hexadecimal number 55?

Solution: 8 bits. Specifically, 0x55 is 01010101 in binary.
**Practice Problem 3.5**

In the screen capture above, what assembly language instruction did the program stop at—i.e., what is the next instruction that will execute, and where in main memory is this instruction stored?

Solution: Note the line "Breakpoint 1 at 0x08048354." This is the address of next instruction to be executed. Looking in the assembler dump, we see that the corresponding instruction is

```
0x08048352 <main+14>: sub esp,eax
0x08048354 <main+16>: mov DWORD PTR [ebp-4],0x7
```

```
0x0804835b <main+23>: mov DWORD PTR [ebp-4],0x7d1
```

**Practice Problem 3.6**

What is the value stored in the eip register? Does this answer make sense?

Solution: Enter the command: `i reip`. You should see that the eip register holds the address 0x8048354. The eip register holds the address of the next instruction that will be executed. In the prior example we saw that the address of the next instruction to be stored is this same value: 0x8048354.

**Practice Problem 3.7**

Refer to the picture shown in Figure 3.1 What should be printed out by each of the following commands? In each case, enter the command to confirm your answer.

(a) `x/xb 0x08048354`

(b) `x/xb 0x08048355`

(c) `x/xb 0x08048356`

(d) `x/xb 0x08048357`

Solution: (a) `c7`  (b) `45`  (c) `fc`  (d) `07`

**Practice Problem 3.8**

What do you think will be displayed by the command: `x/xh 0x08048354`? Confirm your result.

Solution: 0x45c7. This is counterintuitive.

**Practice Problem 3.9**

What do you think will be displayed by the command: `x/xw 0x08048354`? Confirm your result.

Solution: 0x07fc45c7

**Practice Problem 3.10**

What do you think will be displayed by the command: `x/xb $eip`. Confirm your result.

Solution: 0xc7
<table>
<thead>
<tr>
<th>Practice Problem 3.11</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you think will be displayed by the command: \texttt{x/xh $eip}. Confirm your result.</td>
</tr>
<tr>
<td>Solution: 0x45c7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Practice Problem 3.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you think will be displayed by the command: \texttt{x/xw $eip}. Confirm your result.</td>
</tr>
<tr>
<td>Solution: 0x07fc45c7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Practice Problem 3.13</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you think will be displayed by the command: \texttt{x/i $eip}. Confirm your result.</td>
</tr>
<tr>
<td>Solution: \texttt{mov DWORD PTR [esp-4],0x7}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Practice Problem 3.14</th>
</tr>
</thead>
<tbody>
<tr>
<td>When you execute a command (as you just did when you typed \texttt{nexti}), what happens to the instruction pointer (eip)?</td>
</tr>
<tr>
<td>Solution: \texttt{eip} automatically advances to the address of the next instruction.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Practice Problem 3.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the value stored in the \texttt{eip} register? Does this answer make sense?</td>
</tr>
</tbody>
</table>
| Solution: Enter the command: \texttt{i r eip}  
You should see that the \texttt{eip} register holds the address 0x804835b. Yes, this makes sense since it is the address of the next instruction after we entered \texttt{nexti}. |

<table>
<thead>
<tr>
<th>Practice Problem 3.16</th>
</tr>
</thead>
<tbody>
<tr>
<td>What should I type to examine memory to see the integer 7 that has been just placed on the stack? (Confirm your result!)</td>
</tr>
<tr>
<td>Solution: \texttt{x/uw 0xbffff814}, which is the address for \texttt{ebp-4}. Also, \texttt{x/uw $ebp-4}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Practice Problem 3.17</th>
</tr>
</thead>
<tbody>
<tr>
<td>What assembly instruction is located at 0x0804835b?</td>
</tr>
<tr>
<td>Solution: \texttt{mov DWORD PTR [ebp-4],0x7d1}</td>
</tr>
</tbody>
</table>
Practice Problem 3.18

Sketch what you expect the stack to look like after the instruction at address 0x0804835b is executed.

Solution:

```
Practice Problem 3.19

What two things happen when nexti is entered?

Solution: 1. The instruction at which we previously stopped is executed.

2. The eip updates to the next instruction.

Practice Problem 3.20

What should you type to examine memory for the hex values you sketched in Practice Problem 3.18? (Confirm your result!)

Solution: x/x w 0xbffff814 , which is the address for ebp-4. Also, x/xw $ebp-4

Practice Problem 3.21

What should you type to examine memory for the integer 2001? (Confirm your result!)

Solution: x/uw 0xbffff814 , which is the address for ebp-4. Also, x/uw $ebp-4