Recitation Exercises #2

CSCI 2400 Systems
Fall 2014

September 21, 2014

1

Below is the assembly code for a C main function as well as the factorial function which it calls.

```
int factorial(int n)
0:      55                              push   ebp
1:      89 e5                           mov    ebp,esp
3:      83 ec 18                        sub    esp,0x18
6:      83 7d 08 00                     cmp    DWORD PTR [ebp+0x8],0x0
a:      75 07                           jne    13 <factorial+0x13>
c:      b8 01 00 00 00                  mov    eax,0x1
11:     eb 12                           jmp    25 <factorial+0x25>
13:     8b 45 08                        mov    eax,DWORD PTR [ebp+0x8]
16:     83 e8 01                        sub    eax,0x1
19:     89 04 24                        mov    DWORD PTR [esp],eax
1c:     e8 fc ff ff ff                  call   1d <factorial+0x1d>
21:     0f af 45 08                     imul   eax,DWORD PTR [ebp+0x8]
25:     c9                              leave
26:     c3                              ret

int main()
27:     55                              push   ebp
28:     89 e5                           mov    ebp,esp
2a:     83 e4 f0                        and    esp,0xfffffff0
2d:     83 ec 20                        sub    esp,0x20
30:     c7 04 24 07 00 00 00            mov    DWORD PTR [esp],0x7
37:     e8 fc ff ff ff                  call   38 <main+0x11>
3c:     89 44 24 1c                     mov    DWORD PTR [esp+0x1c],eax
40:     8b 44 24 1c                     mov    eax,DWORD PTR [esp+0x1c]
44:     89 44 24 04                     mov    DWORD PTR [esp+0x4],eax
48:     c7 04 24 00 00 00 00            mov    DWORD PTR [esp],0x0
4f:     e8 fc ff ff ff                  call   50 <main+0x29>
54:     b8 00 00 00 00                  mov    eax,0x0
59:     c9                              leave
5a:     c3                              ret
```

1.1

How large is the stack frame created for each call to *factorial*?

1.2

What is the total "distance" covered by the stack pointer from the first call to *factorial* until the end of the last recursive call?

2

Look at the assembly code below and fill out the blanks in the corresponding C code. You need to indicate the data types (short, int, char, etc as well as unsigned if appropriate) for 'a', 'x', 'y', 'q' and 'z', the value to which 'q' is initialized, what is returned and the equivalent computation.

```
function1:
pushl %ebp
movl %esp, %ebp
pushl %edi
pushl %esi
pushl %ebx
subl $16, %esp
movl $43, %esi
movl $0, %ecx
movsbl 8(%ebp),%ebx

.L2:
movzbl %cl, %eax
subw -24(%ebp,%eax,4), %si
leal (%eax,%ebx), %edx
movl %edx, -24(%ebp,%eax,4)
addl $1, %ecx
movzbl %cl, %eax
cmpl 12(%ebp), %eax
jl .L2
movswl %si,%eax
addl %ebx, %eax
addl $16, %esp
popl %ebx
popl %esi
popl %edi
popl %ebp
ret
```

C Code:

```
function1(char a, ________________ x)
{

signed _____ z[3];
______ short y = _____________;
unsgined ___ q = _____________;

do {

________ = _________ - z[__];
___[q] = _________ + ________;
q = ___ + 1;
} while ( _________ < x ); //condition for the "do loop"
return a + y;
}
```