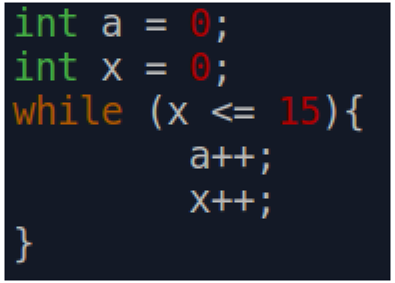
Code Flow 2

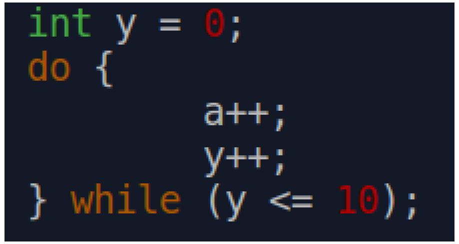
Are we there yet? Are we there yet? Are we there Yet? Are we there yet?
If only we could control all loops...

Loops

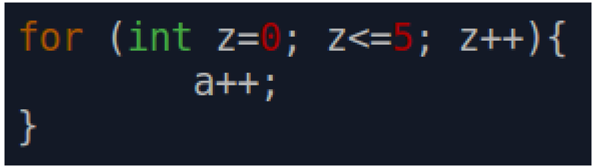
Loops are essential to programming. Technically, you already have the knowledge required to build loops in assembly so let’s just do a quick review.



```
int a = 0;
int x = 0;
while (x <= 15){
        a++;
        x++;
}
```

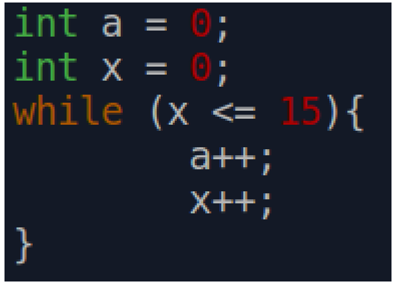


```
int y = 0;
do {
        a++;
        y++;
} while (y <= 10);
```

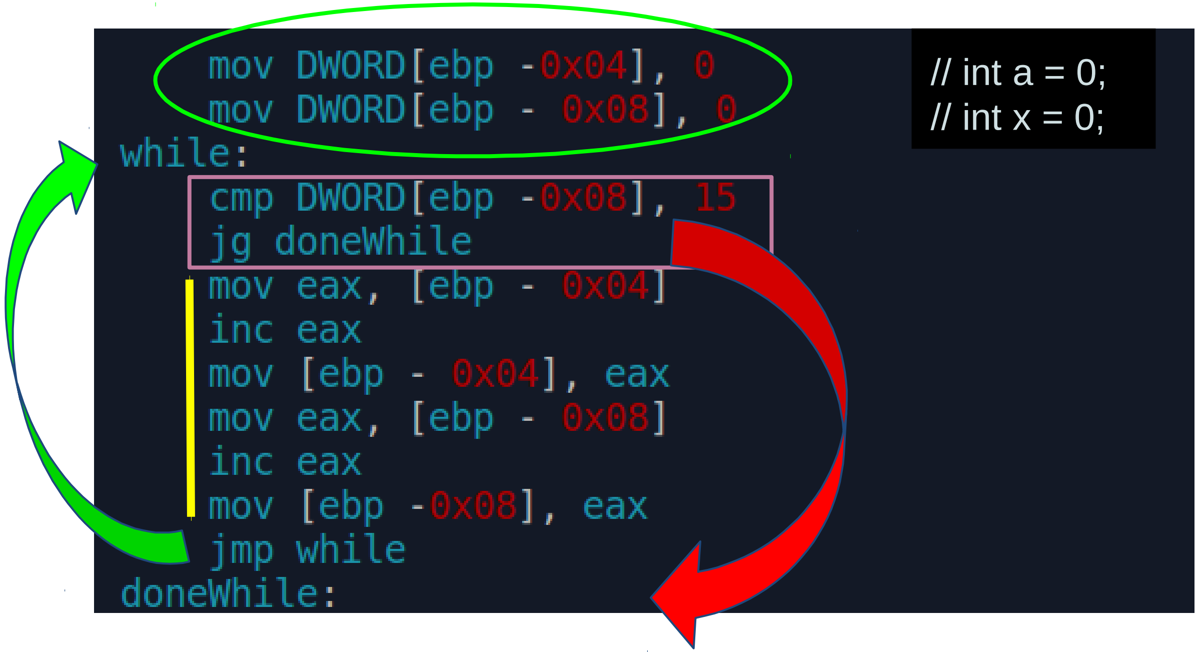


```
for (int z=0; z<=5; z++){
        a++;
}
```

While and For



```
int a = 0;
int x = 0;
while (x <= 15){
        a++;
        x++;
}
```



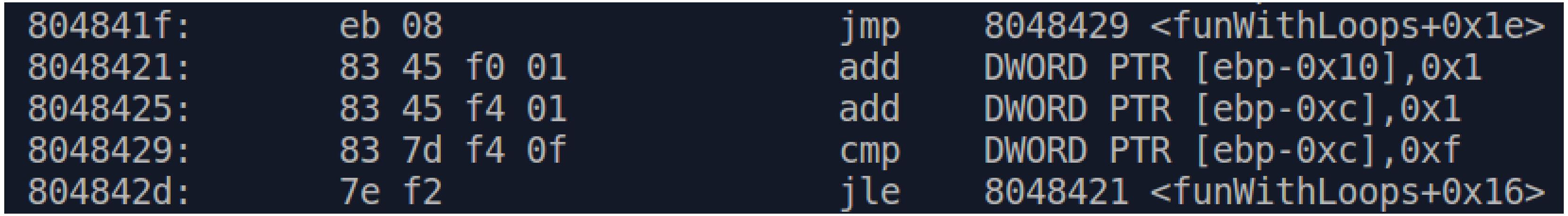
```
    mov DWORD[ebp -0x04], 0
    mov DWORD[ebp - 0x08], 0
while:
    cmp DWORD[ebp -0x08], 15
    jg doneWhile
    mov eax, [ebp - 0x04]
    inc eax
    mov [ebp - 0x04], eax
    mov eax, [ebp - 0x08]
    inc eax
    mov [ebp -0x08], eax
    jmp while
doneWhile:
```

```
// int a = 0;
// int x = 0;
```

As you can see, you do have the required knowledge. This is more a matter of of thinking about the code structure!

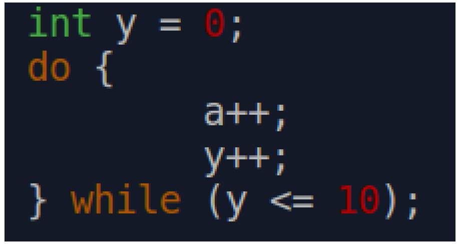
While and For

This form is also frequent and is more desirable. The following code was generated by GCC. It eliminates a conditional jump in the middle of the loop.

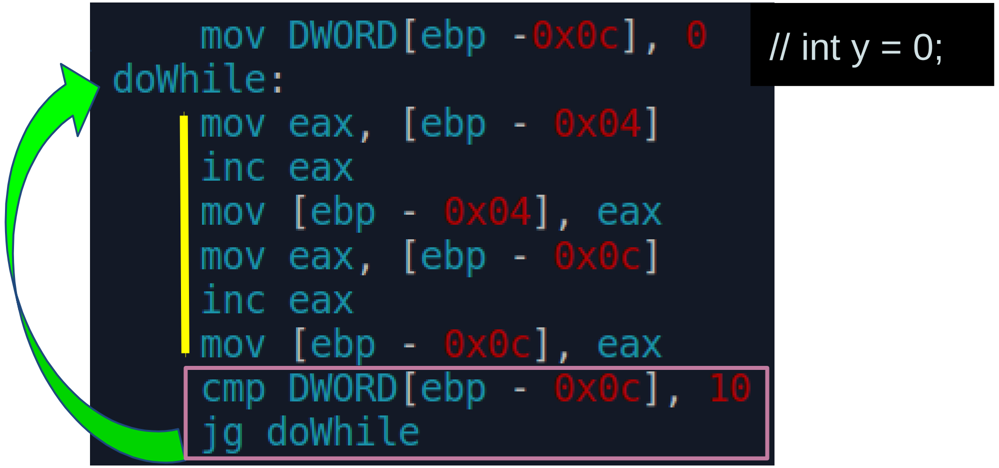


```
804841f:       eb 08                   jmp    8048429 <funWithLoops+0x1e>
8048421:       83 45 f0 01             add    DWORD PTR [ebp-0x10],0x1
8048425:       83 45 f4 01             add    DWORD PTR [ebp-0xc],0x1
8048429:       83 7d f4 0f             cmp    DWORD PTR [ebp-0xc],0xf
804842d:       7e f2                   jle    8048421 <funWithLoops+0x16>
```

Do While



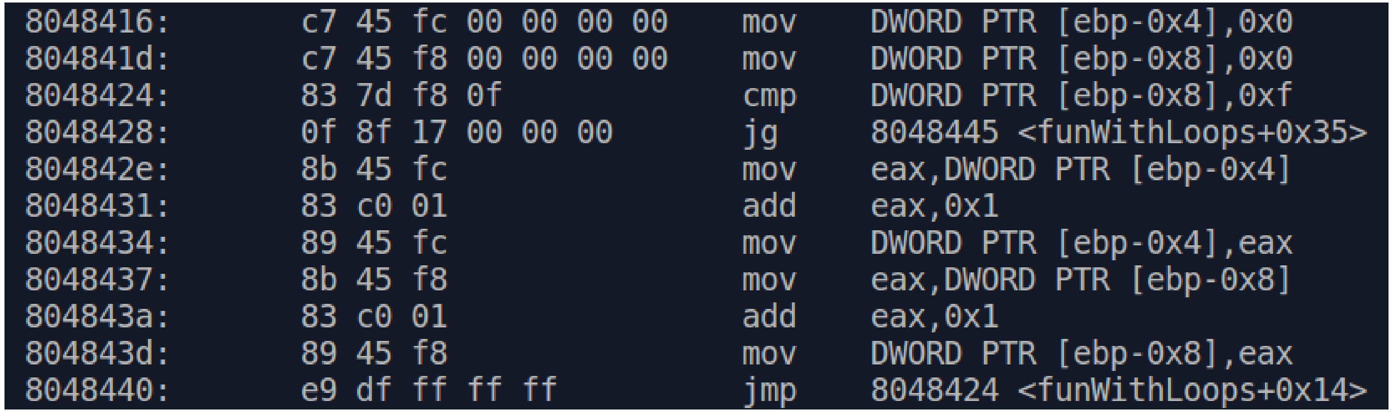
```c
int y = 0;
do {
        a++;
        y++;
} while (y <= 10);
```



```asm
    mov DWORD[ebp -0x0c], 0
doWhile:
    mov eax, [ebp - 0x04]
    inc eax
    mov [ebp - 0x04], eax
    mov eax, [ebp - 0x0c]
    inc eax
    mov [ebp - 0x0c], eax
    cmp DWORD[ebp - 0x0c], 10
    jg doWhile
```

// int y = 0;

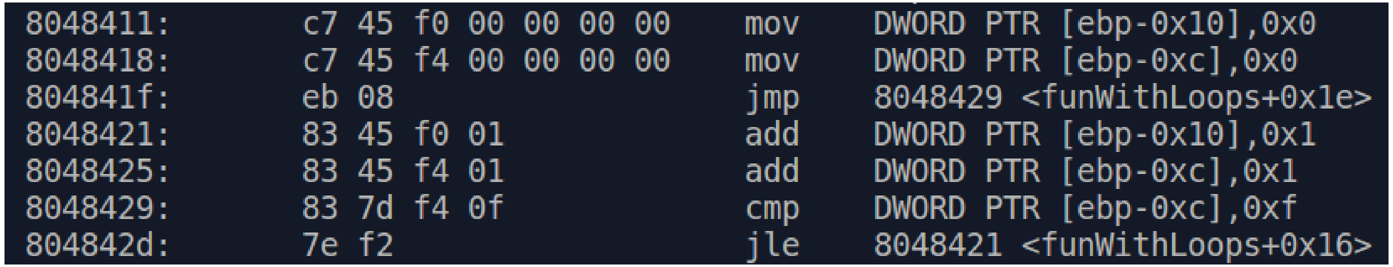
You should be able to see certain advantages in using do while instead of while whenever possible. Can you tell why?



```
8048416:        c7 45 fc 00 00 00 00    mov    DWORD PTR [ebp-0x4],0x0
804841d:        c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
8048424:        83 7d f8 0f             cmp    DWORD PTR [ebp-0x8],0xf
8048428:        0f 8f 17 00 00 00       jg     8048445 <funWithLoops+0x35>
804842e:        8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
8048431:        83 c0 01                add    eax,0x1
8048434:        89 45 fc                mov    DWORD PTR [ebp-0x4],eax
8048437:        8b 45 f8                mov    eax,DWORD PTR [ebp-0x8]
804843a:        83 c0 01                add    eax,0x1
804843d:        89 45 f8                mov    DWORD PTR [ebp-0x8],eax
8048440:        e9 df ff ff ff          jmp    8048424 <funWithLoops+0x14>
```

It all depends on the compiler!

Above: clang, Below: gcc



```
8048411:        c7 45 f0 00 00 00 00    mov    DWORD PTR [ebp-0x10],0x0
8048418:        c7 45 f4 00 00 00 00    mov    DWORD PTR [ebp-0xc],0x0
804841f:        eb 08                   jmp    8048429 <funWithLoops+0x1e>
8048421:        83 45 f0 01             add    DWORD PTR [ebp-0x10],0x1
8048425:        83 45 f4 01             add    DWORD PTR [ebp-0xc],0x1
8048429:        83 7d f4 0f             cmp    DWORD PTR [ebp-0xc],0xf
804842d:        7e f2                   jle    8048421 <funWithLoops+0x16>
```

How about writing some code?