

IF Statements in ASM

```
605 int array[] = {10, 60, 20, 33, 72, 89, 45, 65, 72, 18};
606 int sample = 50;
607 int ArraySize = sizeof array / sizeof sample;
608 int index = 0;
609 int sum = 0;
610
611 while (index < ArraySize) {
612     if (array[index] > sample) {
613         sum += array[index];
614     }
615     index++;
616 }
```

This code calculates the sum of all array elements greater than the value in sample.

The following assembly language code is equivalent to the C++ code above:

```

619 .data
620     sum DWORD 0
621     sample DWORD 50
622     array DWORD 10, 60, 20, 33, 72, 89, 45, 65, 72, 18
623     ArraySize = ($ - Array) / TYPE array
624 .code
625     main PROC
626     mov eax, 0 ; sum
627     mov edx, sample
628     mov esi, 0 ; index
629     mov ecx, ArraySize
630
631     L1:
632     cmp esi, ecx ; if esi < ecx
633     jl L2
634     jmp L5
635
636     L2:
637     cmp array[esi * 4], edx ; if array[esi] > edx
638     jg L3
639     jmp L4
640
641     L3:
642     add eax, array[esi * 4]
643
644     L4:
645     inc esi
646     jmp L1
647
648     L5:
649     mov sum, eax

```

This code works in the following way:

The loop starts by initializing the `eax` register to 0. This register will be used to store the sum of all array elements greater than the value in `sample`. The loop then compares the `esi` register to the `ecx` register.

If the esi register is less than the ecx register, the loop jumps to the L1 label. This means that the loop will continue to iterate until it has examined all of the array elements. If the loop jumps to the L1 label, it then compares the value of the array element at `array[esi * 4]` to the value in the edx register.

If the value of the array element is greater than the value in the edx register, the loop jumps to the L3 label. If the loop jumps to the L3 label, it then adds the value of the array element at `array[esi * 4]` to the eax register.

The loop then increments the esi register and jumps back to the L1 label. This process continues until the loop has examined all of the array elements. Once the loop has examined all of the array elements, it jumps to the L5 label.

This label marks the end of the loop. At the end of the loop, the value of the eax register is stored in the sum variable.

Improvements:

There are a few things that could be done to improve the assembly language code above:

The `cmp` instruction in the L1 label could be replaced with a test instruction.

The test instruction is faster than the `cmp` instruction because it does not set the condition flags. The `jmp` instruction in the L1 label could be replaced with a `loop` instruction.

The `loop` instruction is faster than the `jmp` instruction because it does not push the return address onto the stack. The `cmp` instruction in the L2 label could be replaced with a `sub` instruction.

The `sub` instruction is faster than the `cmp` instruction because it does not set the condition flags. The `jmp` instruction in the L2 label could be replaced with a `jbe` instruction.

The `jbe` instruction is faster than the `jmp` instruction because it does not push the return address onto the stack.

FIGURE 6-1 Loop containing IF statement.



