# **CS 410 Assembly to C++ Activity**

Merrik Wright  
CS410  
Assemly to C++ Information

Step 1: Convert the assembly code into C++ code.

Step 2: Explain the function of the converted C++ code.

| **Assembly Code** | **C++ Code** | **Explanation of Functionality** |
| --- | --- | --- |
| movl −8(%rbp), %eax sall $3, %eax subl $3, %eax movl %eax, −4(%rbp) | int result = (input << 3) - 3; | This code multiplies the variable at -8(%rbp) (likely input) by 8 using a left shift (<< 3) and subtracts 3 from the result. The final value is stored in result, which corresponds to -4(%rbp). |
| movl −8(%rbp), %eax sall $2, %eax subl $1, %eax leal 7(%rax), %edx testl %eax, %eax cmovs %edx, %eax sarl $3, %eax  movl %eax, −4(%rbp) | int temp = (input << 2) - 1;  int result = (temp > 0) ? (temp + 7) >> 3 : temp >> 3; | This code multiplies input by 4 (via left shift) and subtracts 1. If the result is positive, 7 is added before dividing by 8 (for rounding up); otherwise, it's divided by 8 directly. This effectively performs a rounded division: (temp + 7) / 8 for positive numbers, and normal division for negative ones. |
| movl −8(%rbp), %eax leal 7(%rax), %edx testl %eax, %eax cmovs %edx, %eax sarl $3, %eax movl −8(%rbp), %edx sall $2, %edx addl %edx, %eax  movl %eax, −4(%rbp) | int temp = (input > 0) ? (input + 7) >> 3 : input >> 3;  int result = temp + (input << 2); | This code conditionally rounds input up before dividing by 8 if it’s positive (like above), then adds input \* 4 to that value. The result is stored in result. This performs a combination of rounding and scaling operations based on the input’s sign. |