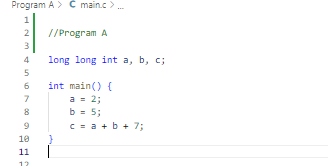
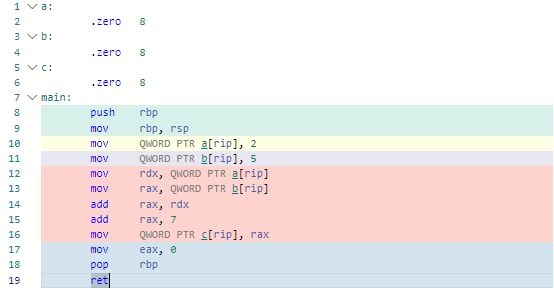
**Program A**

1. Write the C code in ARM DS-5 CE, build the project and then debug it.



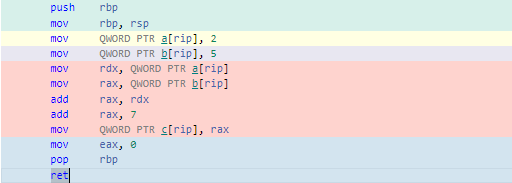
1. ﻿﻿﻿Provide a screenshot of the assembly translation of the code.



1. ﻿﻿﻿Briefly explain the steps (instructions) and registers that were used by the compiler to translate the C source code into Assembly instructions.

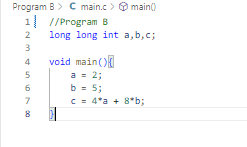
Memory space is reserved for three 8-byte variables, *a*, *b*, and *c*, by using the *.zero 8* directive. The main function starts with a standard prologue that pushes the base pointer onto the stack and establishes a new base pointer. Constants 2 and 5 are placed in memory locations allocated for *a* and *b* respectively. Following this, arithmetic operations are performed by loading the values of a and b into registers rdx and rax. The sum of these values is determined as 7, it is saved in rax. The final outcome is written to the memory location set aside for *c*. The function ends with an epilogue that sets the return value to 0 and restores the base pointer before exiting.

1. ﻿﻿﻿What is the final result of the computed function by the program. Show a screenshot of the register that has the final result after the code is run.

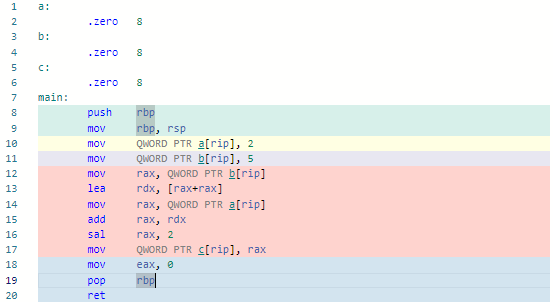


**Program B**

1. Write the C code in ARM DS-5 CE, build the project and then debug it.



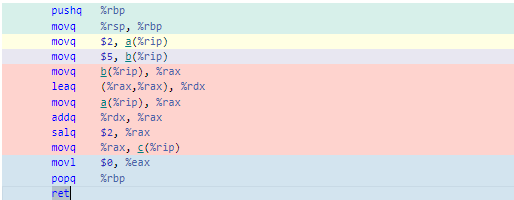
1. ﻿﻿﻿Provide a screenshot of the assembly translation of the code.



1. ﻿﻿﻿Briefly explain the steps (instructions) and registers that were used by the compiler to translate the C source code into Assembly instructions.

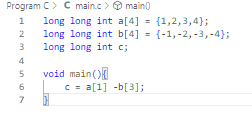
In this program, three 8-byte variables, a, b, and c, are initialized. The main function starts with the standard prologue, pushing the base pointer onto the stack and setting a new base pointer. Values 2 and 5 are then saved in memory locations allocated for a and b, respectively. The computation for variable c, involves loading value b, multiplying it by 2, adding it to value of a and then multiplying the total by 4. The outcome is stored in the memory location designated for c.

1. ﻿﻿﻿What is the final result of the computed function by the program. Show a screenshot of the register that has the final result after the code is run.

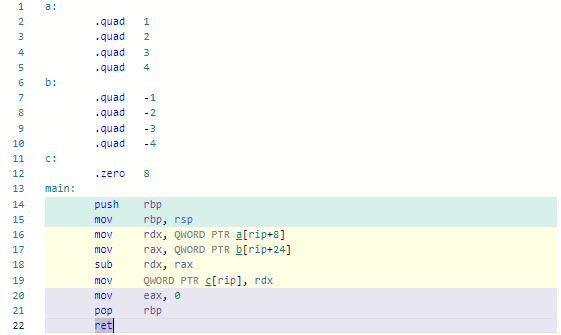


**Program C**

1. Write the C code in ARM DS-5 CE, build the project and then debug it.



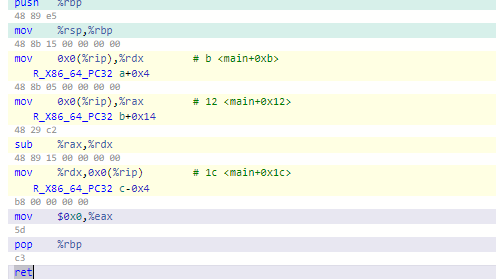
1. ﻿﻿﻿Provide a screenshot of the assembly translation of the code.



1. ﻿﻿﻿Briefly explain the steps (instructions) and registers that were used by the compiler to translate the C source code into Assembly instructions.

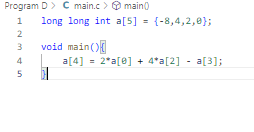
Arrays a and b each have four 8-byte elements. The main function begins by pushing the base pointer onto the stack and establishing a new base pointer. The values in arrays a and b are then loaded into registers rdx and rax. The program proceeds to calculate the difference between the elements of the arrays using the sub instruction, and stores the result in the memory location reserved for variable c. The subtraction operation is performed element-wise by subtracting the corresponding elements of arrays a and b. Finally, the function concludes the return value to 0 and restoring the base pointer before returning.

1. ﻿﻿﻿What is the final result of the computed function by the program. Show a screenshot of the register that has the final result after the code is run.

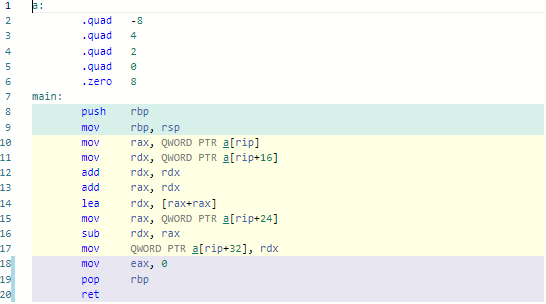


**Program D**

1. Write the C code in ARM DS-5 CE, build the project and then debug it.



1. ﻿﻿﻿Provide a screenshot of the assembly translation of the code.



1. ﻿﻿﻿Briefly explain the steps (instructions) and registers that were used by the compiler to translate the C source code into Assembly instructions.

The program manipulates array a that has five 8-byte elements. The main function starts with pushing the base pointer onto the stack and establishing a new base pointer. The code then proceeds to perform arithmetic operations on the elements of array a. It loads the first and third elements into registers rax and rdx. It doubles the value of the third element, and adds the doubled value to the sum of the first and third elements. The result is stored in rdx. Later, the program loads the fifth element, subtracts it from the computed sum, and stores the final result in the memory location reserved for the last element of array a. The function ends at setting the return value to 0 and restoring the base pointer before returning.

1. ﻿﻿﻿What is the final result of the computed function by the program? Show a screenshot of the register that has the final result after the code is run.

