Binary to C++ with Security Vulnerabilities

CS-410-R4890 Software Reserve Engineering

Raphael Coloma

April 13, 2024

# **CS 410 Binary to C++ With Security Vulnerabilities Activity Template**

**Step 1:** Convert the binary file to assembly code.

**Step 2:** Explain the functionality of the blocks of assembly code.

| **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- |
| Main: | |
| push %rbp  mov %rsp,%rbp | Stack Management for main function. |
| sub $0x20,%rsp | Assign local variables. |
| mov %fs:0x28,%rax  mov %rax,-0x8(%rbp)  xor %eax,%eax | Store sentinel stack-guard value. Exclusive or register %eax by itself to set it to 0. |
| movl $0x0,-0x14(%rbp)  mov -0x14(%rbp),%eax  cmp $0x5,%eax  je 0xd02 <main+655> | Move long value 0 to -0x14 bytes above register %rbp. Move value from -0x14 bytes above register %rbp to register %eax. Compare value 5 to value stored in register %eax. Jump to main+655 if the values are equal |
| lea 0x3a5(%rip),%rsi # 0xe49  lea 0x201575(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x3a5(%rip) into register %rsi. Put memory address 0x201575(%rip) into register %rdi. Print string “----------------\n” to screen. |
| lea 0x3a4(%rip),%rsi # 0xe5b  lea 0x201562(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x3fa4(%rip) into register %rsi. Put memory address 0x201562(%rip) into register %rdi. Print string “- 1)Add -\n” to screen. |
| lea 0x39c(%rip),%rsi # 0xe66  lea 0x20154f(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x39c(%rip) into register %rsi. Put memory address 0x20154f(%rip) into register %rdi. Print string “- 2)Subtract -\n” to screen. |
| lea 0x399(%rip),%rsi # 0xe76  lea 0x20153c(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x399(%rip) into register %rsi. Put memory address 0x20153c(%rip) into register %rdi. Print string “- 3)Multiply -\n” to screen. |
| lea 0x396(%rip),%rsi # 0xe86  lea 0x201529(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x396(%rip) into register %rsi. Put memory address 0x201529(%rip) into register %rdi. Print string “- 4)Exit -\n” to screen. |
| lea 0x346(%rip),%rsi # 0xe49  lea 0x201516(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x346(%rip) into register %rsi. Put memory address 0x201516(%rip) into register %rdi. Print string “----------------\n” to screen. |
| lea -0x14(%rbp),%rax  mov %rax,%rsi  lea 0x201623(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> | Put memory address of 0x14(%rbp) into register %rax.  Move value from register %rax to register %rsi. Put memory address of 0x201623(%rip) into register %rdi. Call cin function for user input. |
| mov -0x14(%rbp),%eax  cmp $0x1,%eax  jne 0xbc3 <main+336> | Move value from 0x14 bytes above register %rbp to register %eax. Compare value 1 to the value in register %eax. Jump to main+336 if values not equal |
| lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201604(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> | Put memory address of 0x10(%rbp) into register %rax.  Move value from register %rax to register %rsi. Put memory address of 0x201604(%rip) into register %rdi. Call cin function for user input. First Value |
| mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 0x870 <\_ZNSirsERi@plt> | Move value from register %rax to register %rdx. Put memory location of -0xc(%rbp) in register %rax. Move value in register %rax to register %rsi. Move value in register %rdx to register %rdi. Call cin function for user input. Second Value |
| mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x2014c1(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from 0x10 bytes above register %rbp to register %eax. Move value from register %eax to register %esi. Put memory address of 0x2014c1(%rip) into register %rdi. Print first user input value to the screen. |
| lea 0x327(%rip),%rsi # 0xe92  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x327(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ - ” to screen. |
| mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 0xc(%rbp) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print second user input value to the screen. |
| lea 0x30c(%rip),%rsi # 0xe96  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x30c(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ = ” to screen. |
| mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  sub %eax,%edx | Move value from register %rax to register %rcx. Move value from 0x10(%rbp) to register %edx. Move value from 0xc(%rbp) to register %eax. Subtract value stored in register %eax from the value stored in register %edx and store the result in register %edx. |
| mov %edx,%eax  mov %eax,%esi  mov %rcx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from register %edx to register %eax. Move value from register %eax to register %esi. Move value from register %rcx to register %rdi. Print result to screen |
| mov %rax,%rdx  mov 0x20141d(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 0x20141d(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character value to screen |
| jmpq 0xa91 <main+30> | While loop condition – jump to main+30 |
| mov -0x14(%rbp),%eax  cmp $0x2,%eax  jne 0xc62 <main+495> | Move the value from 0x14 bytes above %rbp to register %eax. Compare the value 2 to the value stored in register %eax. Jump to main+495 if the values are not equal. |
| lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201563(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> | Put memory address of 0x10(%rbp) into register %rax.  Move value from register %rax to register %rsi. Put memory address of 0x201563(%rip) into register %rdi. Call cin function for user input. First Value |
| mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 0x870 <\_ZNSirsERi@plt> | Move value from register %rax to register %rdx. Put memory location of -0xc(%rbp) in register %rax. Move value in register %rax to register %rsi. Move value in register %rdx to register %rdi. Call cin function for user input. Second Value |
| mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201420(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from 0x10 bytes above register %rbp to register %eax. Move value from register %eax to register %esi. Put memory address of 0x201420(%rip) into register %rdi. Print First user input value to the screen. |
| lea 0x286(%rip),%rsi # 0xe92  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x286(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ - ” to screen. |
| mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 0xc(%rbp) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print second user input value to the screen. |
| lea 0x26b(%rip),%rsi # 0xe96  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x26b(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ = ” to screen. |
| mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  add %edx,%eax | Move value from register %rax to register %rcx. Move value from 0x10(%rbp) to register %edx. Move value from 0xc(%rbp) to register %eax. Add value stored in register %edx from the value stored in register %eax and store the result in register %eax. |
| mov %eax,%esi  mov %rcx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from register %eax to register %esi. Move value from register %rcx to register %rdi. Print result to screen |
| mov %rax,%rdx  mov 0x20137e(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 0x20137e(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character value to screen |
| jmpq 0xa91 <main+30> | While loop condition – jump to main+30 |
| mov -0x14(%rbp),%eax  cmp $0x3,%eax  jne 0xa91 <main+30> | Move the value from 0x14 bytes above %rbp to register %eax. Compare the value 3 to the value stored in register %eax. Jump to main+30 if the values are not equal. |
| lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x2014c4(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> | Put memory address of 0x10(%rbp) into register %rax.  Move value from register %rax to register %rsi. Put memory address of 0x2014c4(%rip) into register %rdi. Call cin function for user input. First Value |
| mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 0x870 <\_ZNSirsERi@plt> | Move value from register %rax to register %rdx. Put memory location of -0xc(%rbp) in register %rax. Move value in register %rax to register %rsi. Move value in register %rdx to register %rdi. Call cin function for user input. Second Value |
| mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201381(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from 0x10 bytes above register %rbp to register %eax. Move value from register %eax to register %esi. Put memory address of 0x201381(%rip) into register %rdi. Print First user input value to the screen. |
| lea 0x1e7(%rip),%rsi # 0xe92  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x1e7(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ - ” to screen. |
| mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 0xc(%rbp) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print second user input value to the screen. |
| lea 0x1cc(%rip),%rsi # 0xe96  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x1cc(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ = ” to screen. |
| mov %rax,%rcx  mov -0x10(%rbp),%eax  mov -0xc(%rbp),%esi  cltd  idiv %esi | Move value from register %rax to register %rcx. Move value from 0x10(%rbp) to register %eax. Move value from 0xc(%rbp) to register %esi. Convert long to double store value in registers edx:eax. Divide value stored in register %esi by the value in register %eax and store the result in register %eax. The remainder is stored in register %edx. |
| mov %eax,%esi  mov %rcx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> | Move value from register %eax to register %esi. Move value from register %rcx to register %rdi. Print result to screen |
| mov %rax,%rdx  mov 0x2012de(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 0x2012de(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character value to screen |
| jmpq 0xa91 <main+30> | While loop condition – jump to main+30 |
| mov $0x0,%eax  mov -0x8(%rbp),%rcx  xor %fs:0x28,%rcx | Move the value 0 to the register %eax. Move the value from 0x8(%rbp) to register %rcx. Exclusive or registers %fs:0x28 and %rcx to check sentinel stack-guard value. |
| je 0xd1b <main+680> | Jumpt to end of main+680 (end of function) if equal |
| callq 0x8b0 <\_\_stack\_chk\_fail@plt> | Call stack\_chk\_fail function - stack protect guard variable has been modified |
| leaveq  retq | Return 0 value from register %eax. Terminate main function. |
| DisplayMenu: | |
| push %rbp  mov %rsp,%rbp | Stack Management for DisplayMenu function |
| lea 0x400(%rip),%rsi # 0xe05  lea 0x201614(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x400(%rip) into register %rsi. Put memory address 0x201614(%rip) into register %rdi. Print string “----------------\n” to screen. |
| lea 0x3fe(%rip),%rsi # 0xe16  lea 0x201601(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x3fe(%rip) into register %rsi. Put memory address 0x201601(%rip) into register %rdi. Print string “- 1)Add -\n” to screen. |
| lea 0x3f5(%rip),%rsi # 0xe20  lea 0x2015ee(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x3f5(%rip) into register %rsi. Put memory address 0x2015ee(%rip) into register %rdi. Print string “- 2)Subtract -\n” to screen. |
| lea 0x3f1(%rip),%rsi # 0xe2f  lea 0x2015db(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x3f1(%rip) into register %rsi. Put memory address 0x2015db(%rip) into register %rdi. Print string “- 3)Multiply -\n” to screen. |
| lea 0x3ed(%rip),%rsi # 0xe3e  lea 0x2015c8(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x3ed(%rip) into register %rsi. Put memory address 0x2015c8(%rip) into register %rdi. Print string “- 4)Exit -\n” to screen. |
| lea 0x3a1(%rip),%rsi # 0xe05  lea 0x2015b5(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address 0x3a1(%rip) into register %rsi. Put memory address 0x2015b5(%rip) into register %rdi. Print string “----------------\n” to screen. |
| nop | No Operation |
| pop %rbp  retq | Restore stack from register %rbp and return to main function. |

**Step 3:** Convert the assembly code to binary.

**Step 4:** Convert the assembly code to C++ code.

#include <iostream>

using namespace std;

void DisplayMenu(){

    cout << "----------------" << endl;

    cout << "- 1)Add -" << endl;

    cout << "- 2)Subtract -" << endl;

    cout << "- 3)Multiply -" << endl;

    cout << "- 4)Exit -" << endl;

    cout << "----------------" << endl;

    return;

}

int main(){

int userInput, firstVal, secondVal, result;

    while(userInput != 5){

        cout << "----------------" << endl;

        cout << "- 1)Add -" << endl;

        cout << "- 2)Subtract -" << endl;

        cout << "- 3)Multiply -" << endl;

        cout << "- 4)Exit -" << endl;

        cout << "----------------" << endl;

        cin >> userInput;

        if(userInput ==1){

            cin >> firstVal;

            cin >> secondVal;

            cout << firstVal << " - " << secondVal << " = ";

            result = firstVal - secondVal;

            cout << result << endl;

        }

        else if(userInput ==2){

            cin >> firstVal;

            cin >> secondVal;

            cout << firstVal << " - " << secondVal << " = ";

            result = firstVal + secondVal;

            cout << result << endl;

        }

        else if(userInput ==3){

            cin >> firstVal;

            cin >> secondVal;

            cout << firstVal << " - " << secondVal << " = ";

            result = firstVal / secondVal;

            cout << result << endl;

        }

    }

return 0;

}

| **Blocks of Assembly Code** | **C++ Code** |
| --- | --- |
| Main: | |
| push %rbp  mov %rsp,%rbp | int main()  { |
| sub $0x20,%rsp | Int userInput, firstVal, secondVal, result; |
| mov %fs:0x28,%rax  mov %rax,-0x8(%rbp)  xor %eax,%eax |
| movl $0x0,-0x14(%rbp)  mov -0x14(%rbp),%eax  cmp $0x5,%eax  je 0xd02 <main+655> | While(userInput != 5)  { |
| lea 0x3a5(%rip),%rsi # 0xe49  lea 0x201575(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “----------------” << endl; |
| lea 0x3a4(%rip),%rsi # 0xe5b  lea 0x201562(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 1)Add -” << endl; |
| lea 0x39c(%rip),%rsi # 0xe66  lea 0x20154f(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 2)Subtract -” << endl; |
| lea 0x399(%rip),%rsi # 0xe76  lea 0x20153c(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 3)Multiply -” << endl; |
| lea 0x396(%rip),%rsi # 0xe86  lea 0x201529(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 4)Exit -” << endl; |
| lea 0x346(%rip),%rsi # 0xe49  lea 0x201516(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “----------------” << endl; |
| lea -0x14(%rbp),%rax  mov %rax,%rsi  lea 0x201623(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> | cin >> userInput; |
| mov -0x14(%rbp),%eax  cmp $0x1,%eax  jne 0xbc3 <main+336> | if(userInput ==1){  cin >> firstVal;  cin >> secondVal;  cout << firstVal << “ - ” << secondVal << “ = ”;  result = firstVal – secondVal;  cout << result << endl;  } |
| lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201604(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> |
| mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 0x870 <\_ZNSirsERi@plt> |
| mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x2014c1(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x8d0 <\_ZNSolsEi@plt> |
| lea 0x327(%rip),%rsi # 0xe92  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |
| mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> |
| lea 0x30c(%rip),%rsi # 0xe96  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |
| mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  sub %eax,%edx |
| mov %edx,%eax  mov %eax,%esi  mov %rcx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> |
| mov %rax,%rdx  mov 0x20141d(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> |
| jmpq 0xa91 <main+30> | …} |
| mov -0x14(%rbp),%eax  cmp $0x2,%eax  jne 0xc62 <main+495> | Else if(userInput ==2){  cin >> firstVal;  cin >> secondVal;  cout << firstVal << “ - ” << secondVal << “ = ”;  result = firstVal + secondVal;  cout << result << endl;  } |
| lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201563(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> |
| mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 0x870 <\_ZNSirsERi@plt> |
| mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201420(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x8d0 <\_ZNSolsEi@plt> |
| lea 0x286(%rip),%rsi # 0xe92  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |
| mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> |
| lea 0x26b(%rip),%rsi # 0xe96  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |
| mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  add %edx,%eax |
| mov %eax,%esi  mov %rcx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> |
| mov %rax,%rdx  mov 0x20137e(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> |
| jmpq 0xa91 <main+30> | …} |
| mov -0x14(%rbp),%eax  cmp $0x3,%eax  jne 0xa91 <main+30> | Else if(userInput ==3){  cin >> firstVal;  cin >> secondVal;  cout << firstVal << “ - ” << secondVal << “ = ”;  result = firstVal / secondVal;  cout << result << endl;  } |
| lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x2014c4(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0x870 <\_ZNSirsERi@plt> |
| mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 0x870 <\_ZNSirsERi@plt> |
| mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201381(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x8d0 <\_ZNSolsEi@plt> |
| lea 0x1e7(%rip),%rsi # 0xe92  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |
| mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> |
| lea 0x1cc(%rip),%rsi # 0xe96  mov %rax,%rdi  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |
| mov %rax,%rcx  mov -0x10(%rbp),%eax  mov -0xc(%rbp),%esi  cltd  idiv %esi |
| mov %eax,%esi  mov %rcx,%rdi  callq 0x8d0 <\_ZNSolsEi@plt> |
| mov %rax,%rdx  mov 0x2012de(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> |
| jmpq 0xa91 <main+30> | …} |
| mov $0x0,%eax  mov -0x8(%rbp),%rcx  xor %fs:0x28,%rcx | return 0;  } |
| je 0xd1b <main+680> |
| callq 0x8b0 <\_\_stack\_chk\_fail@plt> |
| leaveq  retq |
| DisplayMenu: | |
| push %rbp  mov %rsp,%rbp | void DisplayMenu()  { |
| lea 0x400(%rip),%rsi # 0xe05  lea 0x201614(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “----------------” << endl; |
| lea 0x3fe(%rip),%rsi # 0xe16  lea 0x201601(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 1)Add -” << endl; |
| lea 0x3f5(%rip),%rsi # 0xe20  lea 0x2015ee(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 2)Subtract -” << endl; |
| lea 0x3f1(%rip),%rsi # 0xe2f  lea 0x2015db(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 3)Multiply -” << endl; |
| lea 0x3ed(%rip),%rsi # 0xe3e  lea 0x2015c8(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “- 4)Exit -” << endl; |
| lea 0x3a1(%rip),%rsi # 0xe05  lea 0x2015b5(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “----------------” << endl; |
| nop | return;  } |
| pop %rbp  retq |