# **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** |
| --- | --- |
| <+0>: push %rbp  <+1>: mov %rsp,%rbp  <+4>: lea 0x400(%rip),%rsi # 0xe05 '-' <repeats 16 times>  <+11>: lea 0x201614(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+18>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+23>: lea 0x3fe(%rip),%rsi # 0xe16 "- 1)Add -"  <+30>: lea 0x201601(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+37>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+42>: lea 0x3f5(%rip),%rsi # 0xe20 "- 2)Subtract -"  <+49>: lea 0x2015ee(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+56>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+61>: lea 0x3f1(%rip),%rsi # 0xe2f "- 3)Multiply -"  <+68>: lea 0x2015db(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+75>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+80>: lea 0x3ed(%rip),%rsi # 0xe3e "- 4)Exit -"  <+87>: lea 0x2015c8(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+94>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+99>: lea 0x3a1(%rip),%rsi # 0xe05 '-' <repeats 16 times>  <+106>: lea 0x2015b5(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+113>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+118>: nop  <+119>: pop %rbp  <+120>: retq | This block is for the function DisplayMenu. It loads the strings into the registers and then calls cout to display them to the terminal. See below for what the menu looks like.  "----------------\n"  "- 1)Add -\n"  "- 2)Subtract -\n"  "- 3)Multiply -\n"  "- 4)Exit -\n"  "----------------\n" |
| <+0>: push %rbp  <+1>: mov %rsp,%rbp  <+4>: sub $0x20,%rsp  <+8>: mov %fs:0x28,%rax  <+17>: mov %rax,-0x8(%rbp)  <+21>: xor %eax,%eax  <+23>: movl $0x0,-0x14(%rbp)  <+30>: mov -0x14(%rbp),%eax  <+33>: cmp $0x5,%eax  <+36>: je 0xd02 <main+655> | Start of main function and stack is initialized. Value is taken from storage and stored in register. The value is compared to the integer 5 and jumps to the end of the program if true. |
| <+42>: lea 0x3a5(%rip),%rsi # 0xe49 '-' <repeats 16 times>, "\n"  <+49>: lea 0x201575(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+56>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+61>: lea 0x3a4(%rip),%rsi # 0xe5b "- 1)Add -\n"  <+68>: lea 0x201562(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+75>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+80>: lea 0x39c(%rip),%rsi # 0xe66 "- 2)Subtract -\n"  <+87>: lea 0x20154f(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+94>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+99>: lea 0x399(%rip),%rsi # 0xe76 "- 3)Multiply -\n"  <+106>: lea 0x20153c(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+113>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+118>: lea 0x396(%rip),%rsi # 0xe86 "- 4)Exit -\n"  <+125>: lea 0x201529(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+132>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+137>: lea 0x346(%rip),%rsi # 0xe49 '-' <repeats 16 times>, "\n"  <+144>: lea 0x201516(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+151>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | The strings used in the display menu function are moved into registers and displayed in the terminal using cout. |
| <+156>: lea -0x14(%rbp),%rax  <+160>: mov %rax,%rsi  <+163>: lea 0x201623(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+170>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+175>: mov -0x14(%rbp),%eax  <+178>: cmp $0x1,%eax  <+181>: jne 0xbc3 <main+336>  <+187>: lea -0x10(%rbp),%rax  <+191>: mov %rax,%rsi  <+194>: lea 0x201604(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+201>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+206>: mov %rax,%rdx  <+209>: lea -0xc(%rbp),%rax  <+213>: mov %rax,%rsi  <+216>: mov %rdx,%rdi  <+219>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+224>: mov -0x10(%rbp),%eax  <+227>: mov %eax,%esi  <+229>: lea 0x2014c1(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+236>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+241>: lea 0x327(%rip),%rsi # 0xe92 " - "  <+248>: mov %rax,%rdi  <+251>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+256>: mov %rax,%rdx  <+259>: mov -0xc(%rbp),%eax  <+262>: mov %eax,%esi  <+264>: mov %rdx,%rdi  <+267>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+272>: lea 0x30c(%rip),%rsi # 0xe96 " = "  <+279>: mov %rax,%rdi  <+282>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+287>: mov %rax,%rcx  <+290>: mov -0x10(%rbp),%edx  <+293>: mov -0xc(%rbp),%eax  <+296>: sub %eax,%edx  <+298>: mov %edx,%eax  <+300>: mov %eax,%esi  <+302>: mov %rcx,%rdi  <+305>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+310>: mov %rax,%rdx  <+313>: mov 0x20141d(%rip),%rax # 0x201fd0 ""  <+320>: mov %rax,%rsi  <+323>: mov %rdx,%rdi  <+326>: callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> <std::ostream::operator<<(std::ostream& (\*)(std::ostream&))@plt>  <+331>: jmpq 0xa91 <main+30>  <+336>: mov -0x14(%rbp),%eax  <+339>: cmp $0x2,%eax  <+342>: jne 0xc62 <main+495>  <+348>: lea -0x10(%rbp),%rax  <+352>: mov %rax,%rsi  <+355>: lea 0x201563(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+362>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+367>: mov %rax,%rdx  <+370>: lea -0xc(%rbp),%rax  <+374>: mov %rax,%rsi  <+377>: mov %rdx,%rdi  <+380>: callq 0x870 <\_ZNSirsERi@plt>  <+385>: mov -0x10(%rbp),%eax  <+388>: mov %eax,%esi  <+390>: lea 0x201420(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+397>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+402>: lea 0x286(%rip),%rsi # 0xe92 " - "  <+409>: mov %rax,%rdi  <+412>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+417>: mov %rax,%rdx  <+420>: mov -0xc(%rbp),%eax  <+423>: mov %eax,%esi  <+425>: mov %rdx,%rdi  <+428>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+433>: lea 0x26b(%rip),%rsi # 0xe96 " = "  <+440>: mov %rax,%rdi  <+443>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+448>: mov %rax,%rcx  <+451>: mov -0x10(%rbp),%edx  <+454>: mov -0xc(%rbp),%eax  <+457>: add %edx,%eax  <+459>: mov %eax,%esi  <+461>: mov %rcx,%rdi  <+464>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(std::ostream& (\*)(std::ostream&))@plt>  <+469>: mov %rax,%rdx  <+472>: mov 0x20137e(%rip),%rax # 0x201fd0 ""  <+479>: mov %rax,%rsi  <+482>: mov %rdx,%rdi  <+485>: callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt>  <+490>: jmpq 0xa91 <main+30>  <+495>: mov -0x14(%rbp),%eax  <+498>: cmp $0x3,%eax  <+501>: jne 0xa91 <main+30>  <+507>: lea -0x10(%rbp),%rax  <+511>: mov %rax,%rsi  <+514>: lea 0x2014c4(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+521>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+526>: mov %rax,%rdx  <+529>: lea -0xc(%rbp),%rax  <+533>: mov %rax,%rsi  <+536>: mov %rdx,%rdi  <+539>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+544>: mov -0x10(%rbp),%eax  <+547>: mov %eax,%esi  <+549>: lea 0x201381(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+556>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+561>: lea 0x1e7(%rip),%rsi # 0xe92 " - "  <+568>: mov %rax,%rdi  <+571>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+576>: mov %rax,%rdx  <+579>: mov -0xc(%rbp),%eax  <+582>: mov %eax,%esi  <+584>: mov %rdx,%rdi  <+587>: callq 0x8d0 <\_ZNSolsEi@plt> std::ostream::operator<<(int)@plt>  <+592>: lea 0x1cc(%rip),%rsi # 0xe96 " = "  <+599>: mov %rax,%rdi  <+602>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+607>: mov %rax,%rcx  <+610>: mov -0x10(%rbp),%eax  <+613>: mov -0xc(%rbp),%esi  <+616>: cltd  <+617>: idiv %esi  <+619>: mov %eax,%esi  <+621>: mov %rcx,%rdi  <+624>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+629>: mov %rax,%rdx  <+632>: mov 0x2012de(%rip),%rax # 0x201fd0 ""  <+639>: mov %rax,%rsi  <+642>: mov %rdx,%rdi  <+645>: callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> <std::ostream::operator<<(std::ostream& (\*)(std::ostream&))@plt  <+650>: jmpq 0xa91 <main+30> | A value is taken from the user and stored in a register. If the value is 1, 2, or 3 then the program jumps to the respective part of assembly. If the value is neither of those then it jumps to the beginning of the program. Each of the three options takes two numbers and performs arithmetic on them. It will then display the values and strings in the format of # - # = #. The first option subtracts the two numbers, the second adds them, and the third divides them. Once the string has been displayed the program jumps to the beginning of the program. |
| <+655>: mov $0x0,%eax  <+660>: mov -0x8(%rbp),%rcx  <+664>: xor %fs:0x28,%rcx  <+673>: je 0xd1b <main+680>  <+675>: callq 0x8b0 <\_\_stack\_chk\_fail@plt>  <+680>: leaveq  <+681>: retq | The program terminates |

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

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

| **Blocks of Assembly Code** | **C++ Code** |
| --- | --- |
| <+0>: push %rbp  <+1>: mov %rsp,%rbp  <+4>: lea 0x400(%rip),%rsi # 0xe05 '-' <repeats 16 times>  <+11>: lea 0x201614(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+18>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+23>: lea 0x3fe(%rip),%rsi # 0xe16 "- 1)Add -"  <+30>: lea 0x201601(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+37>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+42>: lea 0x3f5(%rip),%rsi # 0xe20 "- 2)Subtract -"  <+49>: lea 0x2015ee(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+56>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+61>: lea 0x3f1(%rip),%rsi # 0xe2f "- 3)Multiply -"  <+68>: lea 0x2015db(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+75>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+80>: lea 0x3ed(%rip),%rsi # 0xe3e "- 4)Exit -"  <+87>: lea 0x2015c8(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+94>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+99>: lea 0x3a1(%rip),%rsi # 0xe05 '-' <repeats 16 times>  <+106>: lea 0x2015b5(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+113>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+118>: nop  <+119>: pop %rbp  <+120>: retq | static void DisplayMenu() {  std::cout << "----------------\n";  std::cout << "- 1)Add -\n";  std::cout << "- 2)Subtract -\n";  std::cout << "- 3)Multiply -\n";  std::cout << "- 4)Exit -\n";  std::cout << "----------------\n";  } |
| <+0>: push %rbp  <+1>: mov %rsp,%rbp  <+4>: sub $0x20,%rsp  <+8>: mov %fs:0x28,%rax  <+17>: mov %rax,-0x8(%rbp)  <+21>: xor %eax,%eax  <+23>: movl $0x0,-0x14(%rbp)  <+30>: mov -0x14(%rbp),%eax  <+33>: cmp $0x5,%eax  <+36>: je 0xd02 <main+655> | int choice = 0, num1, num2, num3;  while (true) {  if (choice == 5) {  return -1;  } |
| <+42>: lea 0x3a5(%rip),%rsi # 0xe49 '-' <repeats 16 times>, "\n"  <+49>: lea 0x201575(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+56>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+61>: lea 0x3a4(%rip),%rsi # 0xe5b "- 1)Add -\n"  <+68>: lea 0x201562(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+75>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+80>: lea 0x39c(%rip),%rsi # 0xe66 "- 2)Subtract -\n"  <+87>: lea 0x20154f(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+94>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+99>: lea 0x399(%rip),%rsi # 0xe76 "- 3)Multiply -\n"  <+106>: lea 0x20153c(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+113>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+118>: lea 0x396(%rip),%rsi # 0xe86 "- 4)Exit -\n"  <+125>: lea 0x201529(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+132>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+137>: lea 0x346(%rip),%rsi # 0xe49 '-' <repeats 16 times>, "\n"  <+144>: lea 0x201516(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+151>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | std::cout << "----------------\n";  std::cout << "- 1)Add -\n";  std::cout << "- 2)Subtract -\n";  std::cout << "- 3)Multiply -\n";  std::cout << "- 4)Exit -\n";  std::cout << "----------------\n"; |
| <+156>: lea -0x14(%rbp),%rax  <+160>: mov %rax,%rsi  <+163>: lea 0x201623(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+170>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+175>: mov -0x14(%rbp),%eax  <+178>: cmp $0x1,%eax  <+181>: jne 0xbc3 <main+336>  <+187>: lea -0x10(%rbp),%rax  <+191>: mov %rax,%rsi  <+194>: lea 0x201604(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+201>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+206>: mov %rax,%rdx  <+209>: lea -0xc(%rbp),%rax  <+213>: mov %rax,%rsi  <+216>: mov %rdx,%rdi  <+219>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+224>: mov -0x10(%rbp),%eax  <+227>: mov %eax,%esi  <+229>: lea 0x2014c1(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+236>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+241>: lea 0x327(%rip),%rsi # 0xe92 " - "  <+248>: mov %rax,%rdi  <+251>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+256>: mov %rax,%rdx  <+259>: mov -0xc(%rbp),%eax  <+262>: mov %eax,%esi  <+264>: mov %rdx,%rdi  <+267>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+272>: lea 0x30c(%rip),%rsi # 0xe96 " = "  <+279>: mov %rax,%rdi  <+282>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+287>: mov %rax,%rcx  <+290>: mov -0x10(%rbp),%edx  <+293>: mov -0xc(%rbp),%eax  <+296>: sub %eax,%edx  <+298>: mov %edx,%eax  <+300>: mov %eax,%esi  <+302>: mov %rcx,%rdi  <+305>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+310>: mov %rax,%rdx  <+313>: mov 0x20141d(%rip),%rax # 0x201fd0 ""  <+320>: mov %rax,%rsi  <+323>: mov %rdx,%rdi  <+326>: callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> <std::ostream::operator<<(std::ostream& (\*)(std::ostream&))@plt>  <+331>: jmpq 0xa91 <main+30>  <+336>: mov -0x14(%rbp),%eax  <+339>: cmp $0x2,%eax  <+342>: jne 0xc62 <main+495>  <+348>: lea -0x10(%rbp),%rax  <+352>: mov %rax,%rsi  <+355>: lea 0x201563(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+362>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+367>: mov %rax,%rdx  <+370>: lea -0xc(%rbp),%rax  <+374>: mov %rax,%rsi  <+377>: mov %rdx,%rdi  <+380>: callq 0x870 <\_ZNSirsERi@plt>  <+385>: mov -0x10(%rbp),%eax  <+388>: mov %eax,%esi  <+390>: lea 0x201420(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+397>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+402>: lea 0x286(%rip),%rsi # 0xe92 " - "  <+409>: mov %rax,%rdi  <+412>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+417>: mov %rax,%rdx  <+420>: mov -0xc(%rbp),%eax  <+423>: mov %eax,%esi  <+425>: mov %rdx,%rdi  <+428>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+433>: lea 0x26b(%rip),%rsi # 0xe96 " = "  <+440>: mov %rax,%rdi  <+443>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+448>: mov %rax,%rcx  <+451>: mov -0x10(%rbp),%edx  <+454>: mov -0xc(%rbp),%eax  <+457>: add %edx,%eax  <+459>: mov %eax,%esi  <+461>: mov %rcx,%rdi  <+464>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(std::ostream& (\*)(std::ostream&))@plt>  <+469>: mov %rax,%rdx  <+472>: mov 0x20137e(%rip),%rax # 0x201fd0 ""  <+479>: mov %rax,%rsi  <+482>: mov %rdx,%rdi  <+485>: callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt>  <+490>: jmpq 0xa91 <main+30>  <+495>: mov -0x14(%rbp),%eax  <+498>: cmp $0x3,%eax  <+501>: jne 0xa91 <main+30>  <+507>: lea -0x10(%rbp),%rax  <+511>: mov %rax,%rsi  <+514>: lea 0x2014c4(%rip),%rdi # 0x202140 <\_ZSt3cin@@GLIBCXX\_3.4>  <+521>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+526>: mov %rax,%rdx  <+529>: lea -0xc(%rbp),%rax  <+533>: mov %rax,%rsi  <+536>: mov %rdx,%rdi  <+539>: callq 0x870 <\_ZNSirsERi@plt> <std::istream::operator>>(int&)@plt>  <+544>: mov -0x10(%rbp),%eax  <+547>: mov %eax,%esi  <+549>: lea 0x201381(%rip),%rdi # 0x202020 <\_ZSt4cout@@GLIBCXX\_3.4>  <+556>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+561>: lea 0x1e7(%rip),%rsi # 0xe92 " - "  <+568>: mov %rax,%rdi  <+571>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+576>: mov %rax,%rdx  <+579>: mov -0xc(%rbp),%eax  <+582>: mov %eax,%esi  <+584>: mov %rdx,%rdi  <+587>: callq 0x8d0 <\_ZNSolsEi@plt> std::ostream::operator<<(int)@plt>  <+592>: lea 0x1cc(%rip),%rsi # 0xe96 " = "  <+599>: mov %rax,%rdi  <+602>: callq 0x890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  <+607>: mov %rax,%rcx  <+610>: mov -0x10(%rbp),%eax  <+613>: mov -0xc(%rbp),%esi  <+616>: cltd  <+617>: idiv %esi  <+619>: mov %eax,%esi  <+621>: mov %rcx,%rdi  <+624>: callq 0x8d0 <\_ZNSolsEi@plt> <std::ostream::operator<<(int)@plt>  <+629>: mov %rax,%rdx  <+632>: mov 0x2012de(%rip),%rax # 0x201fd0 ""  <+639>: mov %rax,%rsi  <+642>: mov %rdx,%rdi  <+645>: callq 0x8a0 <\_ZNSolsEPFRSoS\_E@plt> <std::ostream::operator<<(std::ostream& (\*)(std::ostream&))@plt  <+650>: jmpq 0xa91 <main+30> | std::cin >> choice;  if (choice == 1) {  std::cin >> num1;  std::cin >> num2;  num3 = num1 - num2;  std::cout << num1 << " - " << num2 << " = " << num3 << "\n";  }  else if (choice == 2) {  std::cin >> num1;  std::cin >> num2;  num3 = num1 + num2;  std::cout << num1 << " - " << num2 << " = " << num3 << "\n";  }  else if (choice == 3) {  std::cin >> num1;  std::cin >> num2;  num3 = num1 / num2;  std::cout << num1 << " - " << num2 << " = " << num3 << "\n";  } |
| <+655>: mov $0x0,%eax  <+660>: mov -0x8(%rbp),%rcx  <+664>: xor %fs:0x28,%rcx  <+673>: je 0xd1b <main+680>  <+675>: callq 0x8b0 <\_\_stack\_chk\_fail@plt>  <+680>: leaveq  <+681>: retq | return -1; |