# **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** |
| --- | --- |
| **Function DisplayMenu** |  |
| 0x00005555554009fa <+0>: push %rbp  0x00005555554009fb <+1>: mov %rsp,%rbp | Set up the function |
| 0x00005555554009fe <+4>: lea 0x400(%rip),%rsi # 0x555555400e05  0x0000555555400a05 <+11>: lea 0x201614(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a0c <+18>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a11 <+23>: lea 0x3fe(%rip),%rsi # 0x555555400e16  0x0000555555400a18 <+30>: lea 0x201601(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a1f <+37>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a24 <+42>: lea 0x3f5(%rip),%rsi # 0x555555400e20  0x0000555555400a2b <+49>: lea 0x2015ee(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a32 <+56>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a37 <+61>: lea 0x3f1(%rip),%rsi # 0x555555400e2f  0x0000555555400a3e <+68>: lea 0x2015db(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a45 <+75>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | The entire function is 6 cout statements that print:  ----------------  - 1)Add -  - 2)Subtract -  - 3)Multiply -  - 4)Exit -  ---------------- |
| 0x0000555555400a4a <+80>: lea 0x3ed(%rip),%rsi # 0x555555400e3e  0x0000555555400a51 <+87>: lea 0x2015c8(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a58 <+94>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a5d <+99>: lea 0x3a1(%rip),%rsi # 0x555555400e05  0x0000555555400a64 <+106>: lea 0x2015b5(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a6b <+113>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |  |
| 0x0000555555400a70 <+118>: nop  0x0000555555400a71 <+119>: pop %rbp  0x0000555555400a72 <+120>: retq | Clean up and return from function |
|  |  |
| **Function main** |  |
| 0x0000555555400a73 <+0>: push %rbp  0x0000555555400a74 <+1>: mov %rsp,%rbp  0x0000555555400a77 <+4>: sub $0x20,%rsp  0x0000555555400a7b <+8>: mov %fs:0x28,%rax  0x0000555555400a84 <+17>: mov %rax,-0x8(%rbp)  0x0000555555400a88 <+21>: xor %eax,%eax | Function setup, allocate room for variables and stack protection. |
| 0x0000555555400a8a <+23>: movl $0x0,-0x14(%rbp) | Initialize variable at -0x14(%rbp) to 0 |
| 0x0000555555400a91 <+30>: mov -0x14(%rbp),%eax  0x0000555555400a94 <+33>: cmp $0x5,%eax  0x0000555555400a97 <+36>: je 0x555555400d02 <main+655> | If -0x14(%rbp) is 5 jump to 655 |
| 0x0000555555400a9d <+42>: lea 0x3a5(%rip),%rsi # 0x555555400e49 | This portion of code is the same functionalliy as the DisplayMenu function, but doesn’t use it. |
| 0x0000555555400aa4 <+49>: lea 0x201575(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400aab <+56>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ab0 <+61>: lea 0x3a4(%rip),%rsi # 0x555555400e5b  0x0000555555400ab7 <+68>: lea 0x201562(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400abe <+75>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ac3 <+80>: lea 0x39c(%rip),%rsi # 0x555555400e66  0x0000555555400aca <+87>: lea 0x20154f(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400ad1 <+94>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ad6 <+99>: lea 0x399(%rip),%rsi # 0x555555400e76  0x0000555555400add <+106>: lea 0x20153c(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400ae4 <+113>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ae9 <+118>: lea 0x396(%rip),%rsi # 0x555555400e86    0x0000555555400af0 <+125>: lea 0x201529(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400af7 <+132>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400afc <+137>: lea 0x346(%rip),%rsi # 0x555555400e49  0x0000555555400b03 <+144>: lea 0x201516(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4> | (cont.) |
| 0x0000555555400b0a <+151>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | (cont.) |
| 0x0000555555400b0f <+156>: lea -0x14(%rbp),%rax  0x0000555555400b13 <+160>: mov %rax,%rsi  0x0000555555400b16 <+163>: lea 0x201623(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400b1d <+170>: callq 0x555555400870 <\_ZNSirsERi@plt> | Reads in user input to -0x14(%rbp) form above |
| 0x0000555555400b22 <+175>: mov -0x14(%rbp),%eax  0x0000555555400b25 <+178>: cmp $0x1,%eax  0x0000555555400b28 <+181>: jne 0x555555400bc3 <main+336> | If input is not 1 jump to 336 |
| 0x0000555555400b2e <+187>: lea -0x10(%rbp),%rax  0x0000555555400b32 <+191>: mov %rax,%rsi  0x0000555555400b35 <+194>: lea 0x201604(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400b3c <+201>: callq 0x555555400870 <\_ZNSirsERi@plt>  0x0000555555400b41 <+206>: mov %rax,%rdx | Get user input in variable -0x10(%rbp). |
| 0x0000555555400b44 <+209>: lea -0xc(%rbp),%rax  0x0000555555400b48 <+213>: mov %rax,%rsi  0x0000555555400b4b <+216>: mov %rdx,%rdi  0x0000555555400b4e <+219>: callq 0x555555400870 <\_ZNSirsERi@plt> | Get another user input in variable -0xc(%rbp) |
| 0x0000555555400b53 <+224>: mov -0x10(%rbp),%eax  0x0000555555400b56 <+227>: mov %eax,%esi  0x0000555555400b58 <+229>: lea 0x2014c1(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400b5f <+236>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | Print -0x10(%rbp) |
| 0x0000555555400b64 <+241>: lea 0x327(%rip),%rsi # 0x555555400e92  0x0000555555400b6b <+248>: mov %rax,%rdi  0x0000555555400b6e <+251>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400b73 <+256>: mov %rax,%rdx | Print “ – “ |
| 0x0000555555400b76 <+259>: mov -0xc(%rbp),%eax  0x0000555555400b79 <+262>: mov %eax,%esi  0x0000555555400b7b <+264>: mov %rdx,%rdi  0x0000555555400b7e <+267>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | Print -0xc(%rbp) |
| 0x0000555555400b83 <+272>: lea 0x30c(%rip),%rsi # 0x555555400e96  0x0000555555400b8a <+279>: mov %rax,%rdi  0x0000555555400b8d <+282>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400b92 <+287>: mov %rax,%rcx | Print “ = “ |
| 0x0000555555400b95 <+290>: mov -0x10(%rbp),%edx  0x0000555555400b98 <+293>: mov -0xc(%rbp),%eax  0x0000555555400b9b <+296>: sub %eax,%edx  0x0000555555400b9d <+298>: mov %edx,%eax | Subtract -0x10(%rbp) and -0xc(%rbp) and store result in eax |
| 0x0000555555400b9f <+300>: mov %eax,%esi  0x0000555555400ba1 <+302>: mov %rcx,%rdi  0x0000555555400ba4 <+305>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400ba9 <+310>: mov %rax,%rdx  0x0000555555400bac <+313>: mov 0x20141d(%rip),%rax # 0x555555601fd0  0x0000555555400bb3 <+320>: mov %rax,%rsi  0x0000555555400bb6 <+323>: mov %rdx,%rdi  0x0000555555400bb9 <+326>: callq 0x5555554008a0 <\_ZNSolsEPFRSoS\_E@plt> | Print eax and endl |
| 0x0000555555400bbe <+331>: jmpq 0x555555400a91 <main+30> | Jump back to 30 |
| 0x0000555555400bc3 <+336>: mov -0x14(%rbp),%eax  0x0000555555400bc6 <+339>: cmp $0x2,%eax  0x0000555555400bc9 <+342>: jne 0x555555400c62 <main+495> | If input is not 2 jump to 495 |
| 0x0000555555400bcf <+348>: lea -0x10(%rbp),%rax  0x0000555555400bd3 <+352>: mov %rax,%rsi  0x0000555555400bd6 <+355>: lea 0x201563(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400bdd <+362>: callq 0x555555400870 <\_ZNSirsERi@plt>  0x0000555555400be2 <+367>: mov %rax,%rdx | Get user input in variable -0x10(%rbp). |
| 0x0000555555400be5 <+370>: lea -0xc(%rbp),%rax  0x0000555555400be9 <+374>: mov %rax,%rsi  0x0000555555400bec <+377>: mov %rdx,%rdi  0x0000555555400bef <+380>: callq 0x555555400870 <\_ZNSirsERi@plt> | Get another user input in variable -0xc(%rbp) |
| 0x0000555555400bf4 <+385>: mov -0x10(%rbp),%eax  0x0000555555400bf7 <+388>: mov %eax,%esi  0x0000555555400bf9 <+390>: lea 0x201420(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400c00 <+397>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | Print -0x10(%rbp) |
| 0x0000555555400c05 <+402>: lea 0x286(%rip),%rsi # 0x555555400e92  0x0000555555400c0c <+409>: mov %rax,%rdi  0x0000555555400c0f <+412>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400c14 <+417>: mov %rax,%rdx | Print “ – “ |
| 0x0000555555400c17 <+420>: mov -0xc(%rbp),%eax  0x0000555555400c1a <+423>: mov %eax,%esi  0x0000555555400c1c <+425>: mov %rdx,%rdi  0x0000555555400c1f <+428>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | Print -0xc(%rbp) |
| 0x0000555555400c24 <+433>: lea 0x26b(%rip),%rsi # 0x555555400e96  0x0000555555400c2b <+440>: mov %rax,%rdi  0x0000555555400c2e <+443>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400c33 <+448>: mov %rax,%rcx | Print “ = “ |
| 0x0000555555400c36 <+451>: mov -0x10(%rbp),%edx  0x0000555555400c39 <+454>: mov -0xc(%rbp),%eax  0x0000555555400c3c <+457>: add %edx,%eax | Add -0x10(%rbp) and -0xc(%rbp) and store result in eax |
| 0x0000555555400c3e <+459>: mov %eax,%esi  0x0000555555400c40 <+461>: mov %rcx,%rdi  0x0000555555400c43 <+464>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400c48 <+469>: mov %rax,%rdx  0x0000555555400c4b <+472>: mov 0x20137e(%rip),%rax # 0x555555601fd0  0x0000555555400c52 <+479>: mov %rax,%rsi  0x0000555555400c55 <+482>: mov %rdx,%rdi  0x0000555555400c58 <+485>: callq 0x5555554008a0 <\_ZNSolsEPFRSoS\_E@plt> | Print eax and endl |
| 0x0000555555400c5d <+490>: jmpq 0x555555400a91 <main+30> | Jump to 30 |
| 0x0000555555400c62 <+495>: mov -0x14(%rbp),%eax  0x0000555555400c65 <+498>: cmp $0x3,%eax  0x0000555555400c68 <+501>: jne 0x555555400a91 <main+30> | If input is not 3 jump to 30 |
| 0x0000555555400c6e <+507>: lea -0x10(%rbp),%rax  0x0000555555400c72 <+511>: mov %rax,%rsi  0x0000555555400c75 <+514>: lea 0x2014c4(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400c7c <+521>: callq 0x555555400870 <\_ZNSirsERi@plt>  0x0000555555400c81 <+526>: mov %rax,%rdx | Get user input in variable -0x10(%rbp). |
| 0x0000555555400c84 <+529>: lea -0xc(%rbp),%rax  0x0000555555400c88 <+533>: mov %rax,%rsi  0x0000555555400c8b <+536>: mov %rdx,%rdi  0x0000555555400c8e <+539>: callq 0x555555400870 <\_ZNSirsERi@plt> | Get another user input in variable -0xc(%rbp) |
| 0x0000555555400c93 <+544>: mov -0x10(%rbp),%eax  0x0000555555400c96 <+547>: mov %eax,%esi  0x0000555555400c98 <+549>: lea 0x201381(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400c9f <+556>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | Print -0x10(%rbp) |
| 0x0000555555400ca4 <+561>: lea 0x1e7(%rip),%rsi # 0x555555400e92  0x0000555555400cab <+568>: mov %rax,%rdi  0x0000555555400cae <+571>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400cb3 <+576>: mov %rax,%rdx | Print “ – “ |
| 0x0000555555400cb6 <+579>: mov -0xc(%rbp),%eax  0x0000555555400cb9 <+582>: mov %eax,%esi  0x0000555555400cbb <+584>: mov %rdx,%rdi  0x0000555555400cbe <+587>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | Print -0xc(%rbp) |
| 0x0000555555400cc3 <+592>: lea 0x1cc(%rip),%rsi # 0x555555400e96  0x0000555555400cca <+599>: mov %rax,%rdi  0x0000555555400ccd <+602>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400cd2 <+607>: mov %rax,%rcx | Print “ = “ |
| 0x0000555555400cd5 <+610>: mov -0x10(%rbp),%eax  0x0000555555400cd8 <+613>: mov -0xc(%rbp),%esi  0x0000555555400cdb <+616>: cltd  0x0000555555400cdc <+617>: idiv %esi | Divide -0x10(%rbp) and -0xc(%rbp) and store result in eax |
| 0x0000555555400cde <+619>: mov %eax,%esi  0x0000555555400ce0 <+621>: mov %rcx,%rdi  0x0000555555400ce3 <+624>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400ce8 <+629>: mov %rax,%rdx  0x0000555555400ceb <+632>: mov 0x2012de(%rip),%rax # 0x555555601fd0  0x0000555555400cf2 <+639>: mov %rax,%rsi  0x0000555555400cf5 <+642>: mov %rdx,%rdi  0x0000555555400cf8 <+645>: callq 0x5555554008a0 <\_ZNSolsEPFRSoS\_E@plt> | Print eax and endl |
| 0x0000555555400cfd <+650>: jmpq 0x555555400a91 <main+30> | Jump to 30 |
| 0x0000555555400d02 <+655>: mov $0x0,%eax  0x0000555555400d07 <+660>: mov -0x8(%rbp),%rcx  0x0000555555400d0b <+664>: xor %fs:0x28,%rcx  0x0000555555400d14 <+673>: je 0x555555400d1b <main+680>  0x0000555555400d16 <+675>: callq 0x5555554008b0 <\_\_stack\_chk\_fail@plt>  0x0000555555400d1b <+680>: leaveq  0x0000555555400d1c <+681>: retq | Check for overflow, return 0, clean up and exit function |

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

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

| **Blocks of Assembly Code** | **C++ Code** |
| --- | --- |
| **Function DisplayMenu** |  |
| 0x00005555554009fa <+0>: push %rbp  0x00005555554009fb <+1>: mov %rsp,%rbp | void DisplayMenu { |
| 0x00005555554009fe <+4>: lea 0x400(%rip),%rsi # 0x555555400e05  0x0000555555400a05 <+11>: lea 0x201614(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a0c <+18>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a11 <+23>: lea 0x3fe(%rip),%rsi # 0x555555400e16  0x0000555555400a18 <+30>: lea 0x201601(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a1f <+37>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a24 <+42>: lea 0x3f5(%rip),%rsi # 0x555555400e20  0x0000555555400a2b <+49>: lea 0x2015ee(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a32 <+56>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a37 <+61>: lea 0x3f1(%rip),%rsi # 0x555555400e2f  0x0000555555400a3e <+68>: lea 0x2015db(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a45 <+75>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | cout << “----------------” << endl;  cout << ”- 1)Add –“ << endl;  cout << “- 2)Subtract –“ << endl;  cout << “- 3)Multiply –“ << endl;  cout << “- 4)Exit –“ << endl;  cout << “----------------” << endl; |
| 0x0000555555400a4a <+80>: lea 0x3ed(%rip),%rsi # 0x555555400e3e  0x0000555555400a51 <+87>: lea 0x2015c8(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a58 <+94>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400a5d <+99>: lea 0x3a1(%rip),%rsi # 0x555555400e05  0x0000555555400a64 <+106>: lea 0x2015b5(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400a6b <+113>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> |  |
| 0x0000555555400a70 <+118>: nop  0x0000555555400a71 <+119>: pop %rbp  0x0000555555400a72 <+120>: retq | } |
|  |  |
| **Function main** |  |
| 0x0000555555400a73 <+0>: push %rbp  0x0000555555400a74 <+1>: mov %rsp,%rbp  0x0000555555400a77 <+4>: sub $0x20,%rsp  0x0000555555400a7b <+8>: mov %fs:0x28,%rax  0x0000555555400a84 <+17>: mov %rax,-0x8(%rbp)  0x0000555555400a88 <+21>: xor %eax,%eax | int main() {  int input, num1, num2; |
| 0x0000555555400a8a <+23>: movl $0x0,-0x14(%rbp) | input = 0; |
| 0x0000555555400a91 <+30>: mov -0x14(%rbp),%eax  0x0000555555400a94 <+33>: cmp $0x5,%eax  0x0000555555400a97 <+36>: je 0x555555400d02 <main+655> | while (input != 5) { |
| 0x0000555555400a9d <+42>: lea 0x3a5(%rip),%rsi # 0x555555400e49 | (next page) |
| 0x0000555555400aa4 <+49>: lea 0x201575(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400aab <+56>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ab0 <+61>: lea 0x3a4(%rip),%rsi # 0x555555400e5b  0x0000555555400ab7 <+68>: lea 0x201562(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400abe <+75>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ac3 <+80>: lea 0x39c(%rip),%rsi # 0x555555400e66  0x0000555555400aca <+87>: lea 0x20154f(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400ad1 <+94>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ad6 <+99>: lea 0x399(%rip),%rsi # 0x555555400e76  0x0000555555400add <+106>: lea 0x20153c(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400ae4 <+113>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ae9 <+118>: lea 0x396(%rip),%rsi # 0x555555400e86    0x0000555555400af0 <+125>: lea 0x201529(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400af7 <+132>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400afc <+137>: lea 0x346(%rip),%rsi # 0x555555400e49  0x0000555555400b03 <+144>: lea 0x201516(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4> | cout << “----------------” << endl;  cout << ”- 1)Add –“ << endl;  cout << “- 2)Subtract –“ << endl;  cout << “- 3)Multiply –“ << endl;  cout << “- 4)Exit –“ << endl;  cout << “----------------” << endl; |
| 0x0000555555400b0a <+151>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | (cont.) |
| 0x0000555555400b0f <+156>: lea -0x14(%rbp),%rax  0x0000555555400b13 <+160>: mov %rax,%rsi  0x0000555555400b16 <+163>: lea 0x201623(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400b1d <+170>: callq 0x555555400870 <\_ZNSirsERi@plt> | cin >> input; |
| 0x0000555555400b22 <+175>: mov -0x14(%rbp),%eax  0x0000555555400b25 <+178>: cmp $0x1,%eax  0x0000555555400b28 <+181>: jne 0x555555400bc3 <main+336> | switch (input) {  case 1: |
| 0x0000555555400b2e <+187>: lea -0x10(%rbp),%rax  0x0000555555400b32 <+191>: mov %rax,%rsi  0x0000555555400b35 <+194>: lea 0x201604(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400b3c <+201>: callq 0x555555400870 <\_ZNSirsERi@plt>  0x0000555555400b41 <+206>: mov %rax,%rdx | cin >> num1; |
| 0x0000555555400b44 <+209>: lea -0xc(%rbp),%rax  0x0000555555400b48 <+213>: mov %rax,%rsi  0x0000555555400b4b <+216>: mov %rdx,%rdi  0x0000555555400b4e <+219>: callq 0x555555400870 <\_ZNSirsERi@plt> | cin >> num2; |
| 0x0000555555400b53 <+224>: mov -0x10(%rbp),%eax  0x0000555555400b56 <+227>: mov %eax,%esi  0x0000555555400b58 <+229>: lea 0x2014c1(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400b5f <+236>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | cout << num1 << “ – “ << num2 << “ = “ << num1 – num2 << endl; |
| 0x0000555555400b64 <+241>: lea 0x327(%rip),%rsi # 0x555555400e92  0x0000555555400b6b <+248>: mov %rax,%rdi  0x0000555555400b6e <+251>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400b73 <+256>: mov %rax,%rdx  0x0000555555400b76 <+259>: mov -0xc(%rbp),%eax  0x0000555555400b79 <+262>: mov %eax,%esi  0x0000555555400b7b <+264>: mov %rdx,%rdi  0x0000555555400b7e <+267>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400b83 <+272>: lea 0x30c(%rip),%rsi # 0x555555400e96  0x0000555555400b8a <+279>: mov %rax,%rdi  0x0000555555400b8d <+282>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400b92 <+287>: mov %rax,%rcx  0x0000555555400b95 <+290>: mov -0x10(%rbp),%edx  0x0000555555400b98 <+293>: mov -0xc(%rbp),%eax  0x0000555555400b9b <+296>: sub %eax,%edx  0x0000555555400b9d <+298>: mov %edx,%eax  0x0000555555400b9f <+300>: mov %eax,%esi  0x0000555555400ba1 <+302>: mov %rcx,%rdi  0x0000555555400ba4 <+305>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400ba9 <+310>: mov %rax,%rdx | (cont.) |
| 0x0000555555400bac <+313>: mov 0x20141d(%rip),%rax # 0x555555601fd0  0x0000555555400bb3 <+320>: mov %rax,%rsi  0x0000555555400bb6 <+323>: mov %rdx,%rdi  0x0000555555400bb9 <+326>: callq 0x5555554008a0 <\_ZNSolsEPFRSoS\_E@plt> | (cont.) |
| 0x0000555555400bbe <+331>: jmpq 0x555555400a91 <main+30> | break; |
| 0x0000555555400bc3 <+336>: mov -0x14(%rbp),%eax  0x0000555555400bc6 <+339>: cmp $0x2,%eax  0x0000555555400bc9 <+342>: jne 0x555555400c62 <main+495> | case 2: |
| 0x0000555555400bcf <+348>: lea -0x10(%rbp),%rax  0x0000555555400bd3 <+352>: mov %rax,%rsi  0x0000555555400bd6 <+355>: lea 0x201563(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400bdd <+362>: callq 0x555555400870 <\_ZNSirsERi@plt>  0x0000555555400be2 <+367>: mov %rax,%rdx | cin >> num1; |
| 0x0000555555400be5 <+370>: lea -0xc(%rbp),%rax  0x0000555555400be9 <+374>: mov %rax,%rsi  0x0000555555400bec <+377>: mov %rdx,%rdi  0x0000555555400bef <+380>: callq 0x555555400870 <\_ZNSirsERi@plt> | cin >> num2; |
| 0x0000555555400bf4 <+385>: mov -0x10(%rbp),%eax  0x0000555555400bf7 <+388>: mov %eax,%esi  0x0000555555400bf9 <+390>: lea 0x201420(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400c00 <+397>: callq 0x5555554008d0 <\_ZNSolsEi@plt> | cout << num1 << “ – “ << num2 << “ = “ << num1 + num2 << endl; |
| 0x0000555555400c05 <+402>: lea 0x286(%rip),%rsi # 0x555555400e92  0x0000555555400c0c <+409>: mov %rax,%rdi  0x0000555555400c0f <+412>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400c14 <+417>: mov %rax,%rdx  0x0000555555400c17 <+420>: mov -0xc(%rbp),%eax  0x0000555555400c1a <+423>: mov %eax,%esi  0x0000555555400c1c <+425>: mov %rdx,%rdi  0x0000555555400c1f <+428>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400c24 <+433>: lea 0x26b(%rip),%rsi # 0x555555400e96  0x0000555555400c2b <+440>: mov %rax,%rdi  0x0000555555400c2e <+443>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400c33 <+448>: mov %rax,%rcx  0x0000555555400c36 <+451>: mov -0x10(%rbp),%edx  0x0000555555400c39 <+454>: mov -0xc(%rbp),%eax  0x0000555555400c3c <+457>: add %edx,%eax  0x0000555555400c3e <+459>: mov %eax,%esi  0x0000555555400c40 <+461>: mov %rcx,%rdi  0x0000555555400c43 <+464>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400c48 <+469>: mov %rax,%rdx  0x0000555555400c4b <+472>: mov 0x20137e(%rip),%rax # 0x555555601fd0  0x0000555555400c52 <+479>: mov %rax,%rsi  0x0000555555400c55 <+482>: mov %rdx,%rdi  0x0000555555400c58 <+485>: callq 0x5555554008a0 <\_ZNSolsEPFRSoS\_E@plt> | (cont.) |
| 0x0000555555400c5d <+490>: jmpq 0x555555400a91 <main+30> | break; |
| 0x0000555555400c62 <+495>: mov -0x14(%rbp),%eax  0x0000555555400c65 <+498>: cmp $0x3,%eax  0x0000555555400c68 <+501>: jne 0x555555400a91 <main+30> | case 3: |
| 0x0000555555400c6e <+507>: lea -0x10(%rbp),%rax  0x0000555555400c72 <+511>: mov %rax,%rsi  0x0000555555400c75 <+514>: lea 0x2014c4(%rip),%rdi # 0x555555602140 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400c7c <+521>: callq 0x555555400870 <\_ZNSirsERi@plt>  0x0000555555400c81 <+526>: mov %rax,%rdx | cin >> num1; |
| 0x0000555555400c84 <+529>: lea -0xc(%rbp),%rax  0x0000555555400c88 <+533>: mov %rax,%rsi  0x0000555555400c8b <+536>: mov %rdx,%rdi  0x0000555555400c8e <+539>: callq 0x555555400870 <\_ZNSirsERi@plt> | cin >> num2; |
| 0x0000555555400c93 <+544>: mov -0x10(%rbp),%eax  0x0000555555400c96 <+547>: mov %eax,%esi  0x0000555555400c98 <+549>: lea 0x201381(%rip),%rdi # 0x555555602020 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400c9f <+556>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400ca4 <+561>: lea 0x1e7(%rip),%rsi # 0x555555400e92  0x0000555555400cab <+568>: mov %rax,%rdi  0x0000555555400cae <+571>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400cb3 <+576>: mov %rax,%rdx | cout << num1 << “ – “ num2 << “ = “ << num1 / num2 << endl; |
| 0x0000555555400cb6 <+579>: mov -0xc(%rbp),%eax  0x0000555555400cb9 <+582>: mov %eax,%esi  0x0000555555400cbb <+584>: mov %rdx,%rdi  0x0000555555400cbe <+587>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400cc3 <+592>: lea 0x1cc(%rip),%rsi # 0x555555400e96  0x0000555555400cca <+599>: mov %rax,%rdi  0x0000555555400ccd <+602>: callq 0x555555400890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400cd2 <+607>: mov %rax,%rcx  0x0000555555400cd5 <+610>: mov -0x10(%rbp),%eax  0x0000555555400cd8 <+613>: mov -0xc(%rbp),%esi  0x0000555555400cdb <+616>: cltd  0x0000555555400cdc <+617>: idiv %esi  0x0000555555400cde <+619>: mov %eax,%esi  0x0000555555400ce0 <+621>: mov %rcx,%rdi  0x0000555555400ce3 <+624>: callq 0x5555554008d0 <\_ZNSolsEi@plt>  0x0000555555400ce8 <+629>: mov %rax,%rdx  0x0000555555400ceb <+632>: mov 0x2012de(%rip),%rax # 0x555555601fd0  0x0000555555400cf2 <+639>: mov %rax,%rsi  0x0000555555400cf5 <+642>: mov %rdx,%rdi  0x0000555555400cf8 <+645>: callq 0x5555554008a0 <\_ZNSolsEPFRSoS\_E@plt> | (cont.) |
| 0x0000555555400cfd <+650>: jmpq 0x555555400a91 <main+30> | break;  } |
| 0x0000555555400d02 <+655>: mov $0x0,%eax  0x0000555555400d07 <+660>: mov -0x8(%rbp),%rcx  0x0000555555400d0b <+664>: xor %fs:0x28,%rcx  0x0000555555400d14 <+673>: je 0x555555400d1b <main+680>  0x0000555555400d16 <+675>: callq 0x5555554008b0 <\_\_stack\_chk\_fail@plt>  0x0000555555400d1b <+680>: leaveq  0x0000555555400d1c <+681>: retq | return 0;  } |