# CS 410 Project One Proficiency Test Template

## Explain the functionality of the blocks of assembly code.

### “main” function”

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 0x0000555555400e4a <+0>: push %rbp  0x0000555555400e4b <+1>: mov %rsp,%rbp | Setup main function |
| 0x0000555555400e4e <+4>: lea 0x5eb(%rip),%rsi # 0x555555401440  0x0000555555400e55 <+11>: lea 0x201244(%rip),%rdi # 0x5555556020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400e5c <+18>: callq 0x555555400c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Print string to console  “Hello! Welcome to our Investment Company” |
| 0x0000555555400e61 <+23>: callq 0x555555400f6a <\_Z25CheckUserPermissionAccessv> | Call function CheckUserPermissionAccess |
| 0x0000555555400e66 <+28>: mov %eax,0x201494(%rip) # 0x555555602300 <answer> | Move return value of CheckUserPermissionAccess into 0x201494(%rip) (answer). I think this is a global variable since we didn’t allocate any memory for local variables. |
| 0x0000555555400e6c <+34>: mov 0x20148e(%rip),%eax # 0x555555602300 <answer>  0x0000555555400e72 <+40>: cmp $0x1,%eax  0x0000555555400e75 <+43>: je 0x555555400e8a <main+64> | If answer is equal to 1 jump to 64. |
| 0x0000555555400e77 <+45>: lea 0x5f2(%rip),%rsi # 0x555555401470  0x0000555555400e7e <+52>: lea 0x20121b(%rip),%rdi # 0x5555556020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400e85 <+59>: callq 0x555555400c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Print string to console  “Invalid Password. Please try again” |
| 0x0000555555400e8a <+64>: mov 0x201470(%rip),%eax # 0x555555602300 <answer>  0x0000555555400e90 <+70>: cmp $0x1,%eax  0x0000555555400e93 <+73>: je 0x555555400e97 <main+77> | If answer is equal to 1 jump to 77. |
| 0x0000555555400e95 <+75>: jmp 0x555555400e61 <main+23> | Jump back to 23. Basically if CheckUserPermissionAccess is not 1 this will happen |
| 0x0000555555400e97 <+77>: lea 0x5f6(%rip),%rsi # 0x555555401494  0x0000555555400e9e <+84>: lea 0x2011fb(%rip),%rdi # 0x5555556020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400ea5 <+91>: callq 0x555555400c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400eaa <+96>: lea 0x5ff(%rip),%rsi # 0x5555554014b0  0x0000555555400eb1 <+103>: lea 0x2011e8(%rip),%rdi # 0x5555556020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400eb8 <+110>: callq 0x555555400c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ebd <+115>: lea 0x614(%rip),%rsi # 0x5555554014d8  0x0000555555400ec4 <+122>: lea 0x2011d5(%rip),%rdi # 0x5555556020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400ecb <+129>: callq 0x555555400c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400ed0 <+134>: lea 0x625(%rip),%rsi # 0x5555554014fc  0x0000555555400ed7 <+141>: lea 0x2011c2(%rip),%rdi # 0x5555556020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400ede <+148>: callq 0x555555400c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Print several strings to console  “What would you like to do?”  “DISPLAY the client list (enter 1)”  “CHANGE a client's choice (enter 2)”  “Exit the program.. (enter 3)” |
| 0x0000555555400ee3 <+153>: lea 0x20140a(%rip),%rsi # 0x5555556022f4 <choice>  0x0000555555400eea <+160>: lea 0x2012cf(%rip),%rdi # 0x5555556021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000555555400ef1 <+167>: callq 0x555555400c60 <\_ZNSirsERi@plt> | Get input into global variable choice. |
| 0x0000555555400ef6 <+172>: lea 0x61d(%rip),%rsi # 0x55555540151a  0x0000555555400efd <+179>: lea 0x20119c(%rip),%rdi # 0x5555556020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000555555400f04 <+186>: callq 0x555555400c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000555555400f09 <+191>: mov %rax,%rdx  0x0000555555400f0c <+194>: mov 0x2013e2(%rip),%eax # 0x5555556022f4 <choice>  0x0000555555400f12 <+200>: mov %eax,%esi  0x0000555555400f14 <+202>: mov %rdx,%rdi  0x0000555555400f17 <+205>: callq 0x555555400d00 <\_ZNSolsEi@plt>  0x0000555555400f1c <+210>: mov %rax,%rd 0x0000555555400f1f <+213>: mov 0x2010aa(%rip),%rax # 0x555555601fd0  0x0000555555400f26 <+220>: mov %rax,%rsi  0x0000555555400f29 <+223>: mov %rdx,%rdi  0x0000555555400f2c <+226>: callq 0x555555400ca0 <\_ZNSolsEPFRSoS\_E@plt> | Print strings to console.  “You chose “ << choice << endl; |
| 0x0000555555400f31 <+231>: mov 0x2013bd(%rip),%eax # 0x5555556022f4 <choice>  0x0000555555400f37 <+237>: cmp $0x1,%eax  0x0000555555400f3a <+240>: jne 0x555555400f43 <main+249> | If choice not equal to 1 jump to 249 |
| 0x0000555555400f3c <+242>: callq 0x55555540108b <\_Z11DisplayInfov>  0x0000555555400f41 <+247>: jmp 0x555555400f53 <main+265> | Call function DisplayInfo. This is done when choice is 1. Then jump to 265 |
| 0x0000555555400f43 <+249>: mov 0x2013ab(%rip),%eax # 0x5555556022f4 <choice>  0x0000555555400f49 <+255>: cmp $0x2,%eax  0x0000555555400f4c <+258>: jne 0x555555400f53 <main+265> | If choice not equal to 2 jump to 265 |
| 0x0000555555400f4e <+260>: callq 0x555555401277 <\_Z20ChangeCustomerChoicev> | Call function ChangeCustomerChoice. This is done when choice is 2. |
| 0x0000555555400f53 <+265>: mov 0x20139b(%rip),%eax # 0x5555556022f4 <choice>  0x0000555555400f59 <+271>: cmp $0x3,%eax  0x0000555555400f5c <+274>: je 0x555555400f63 <main+281> | If choice is equal to 3 jump to 281 |
| 0x0000555555400f5e <+276>: jmpq 0x555555400e97 <main+77> | Jump to 77. This is done if choice is not 3. |
| 0x0000555555400f63 <+281>: mov $0x0,%eax  0x0000555555400f68 <+286>: pop %rbp  0x0000555555400f69 <+287>: retq | End main function and return 0. |

### ChangeCustomerChoice function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 0x0000000000001277 <+0>: push %rbp  0x0000000000001278 <+1>: mov %rsp,%rbp | Set up the function |
| 0x000000000000127b <+4>: lea 0x346(%rip),%rsi # 0x15c8  0x0000000000001282 <+11>: lea 0x200e17(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000000000001289 <+18>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Print to console  “Enter the number of the client that you wish to change” |
| 0x000000000000128e <+23>: lea 0x201063(%rip),%rsi # 0x2022f8 <changechoice>  0x0000000000001295 <+30>: lea 0x200f24(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  0x000000000000129c <+37>: callq 0xc60 <\_ZNSirsERi@plt> | Get user input for global variable changechoice |
| 0x00000000000012a1 <+42>: lea 0x358(%rip),%rsi # 0x1600  0x00000000000012a8 <+49>: lea 0x200df1(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x00000000000012af <+56>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Print to console  “Please enter the client's new service choice (1 = Brokerage, 2 = Retirement)” |
| 0x00000000000012b4 <+61>: lea 0x201041(%rip),%rsi # 0x2022fc <newservice>  0x00000000000012bb <+68>: lea 0x200efe(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  0x00000000000012c2 <+75>: callq 0xc60 <\_ZNSirsERi@plt> | Get user input for global variable newservice. |
| 0x00000000000012c7 <+80>: mov 0x20102b(%rip),%eax # 0x2022f8 <changechoice>  0x00000000000012cd <+86>: cmp $0x1,%eax  0x00000000000012d0 <+89>: jne 0x12e0 <\_Z20ChangeCustomerChoicev+105> | If changechoice is not 1 jump to 105 |
| 0x00000000000012d2 <+91>: mov 0x201024(%rip),%eax # 0x2022fc <newservice>  0x00000000000012d8 <+97>: mov %eax,0x200d82(%rip) # 0x202060 <num1>  0x00000000000012de <+103>: jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Sets global variable num1 to newservice then jump to 203 |
| 0x00000000000012e0 <+105>: mov 0x201012(%rip),%eax # 0x2022f8 <changechoice>  0x00000000000012e6 <+111>: cmp $0x2,%eax  0x00000000000012e9 <+114>: jne 0x12f9 <\_Z20ChangeCustomerChoicev+130> | If changechoice is not 2 jump to 130 |
| 0x00000000000012eb <+116>: mov 0x20100b(%rip),%eax # 0x2022fc <newservice>  0x00000000000012f1 <+122>: mov %eax,0x200d6d(%rip) # 0x202064 <num2>  0x00000000000012f7 <+128>: jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Sets global variable num2 to newservice then jump to 203 |
| 0x00000000000012f9 <+130>: mov 0x200ff9(%rip),%eax # 0x2022f8 <changechoice>  0x00000000000012ff <+136>: cmp $0x3,%eax  0x0000000000001302 <+139>: jne 0x1312 <\_Z20ChangeCustomerChoicev+155> | If changechoice is not 3 jump to 155 |
| 0x0000000000001304 <+141>: mov 0x200ff2(%rip),%eax # 0x2022fc <newservice>  0x000000000000130a <+147>: mov %eax,0x200d58(%rip) # 0x202068 <num3>  0x0000000000001310 <+153>: jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Sets global variable num3 to newservice then jump to 203 |
| 0x0000000000001312 <+155>: mov 0x200fe0(%rip),%eax # 0x2022f8 <changechoice>  0x0000000000001318 <+161>: cmp $0x4,%eax  0x000000000000131b <+164>: jne 0x132b <\_Z20ChangeCustomerChoicev+180> | If changechoice is not 4 jump to 180 |
| 0x000000000000131d <+166>: mov 0x200fd9(%rip),%eax # 0x2022fc <newservice>  0x0000000000001323 <+172>: mov %eax,0x200d43(%rip) # 0x20206c <num4>  0x0000000000001329 <+178>: jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Sets global variable num4 to newservice then jump to 203 |
| 0x000000000000132b <+180>: mov 0x200fc7(%rip),%eax # 0x2022f8 <changechoice>  0x0000000000001331 <+186>: cmp $0x5,%eax  0x0000000000001334 <+189>: jne 0x1342 <\_Z20ChangeCustomerChoicev+203> | If changechoice is not 5 jump to 203 |
| 0x0000000000001336 <+191>: mov 0x200fc0(%rip),%eax # 0x2022fc <newservice>  0x000000000000133c <+197>: mov %eax,0x200d2e(%rip) # 0x202070 <num5> | Sets global variable num5 to newservice |
| 0x0000000000001342 <+203>: nop  0x0000000000001343 <+204>: pop %rbp  0x0000000000001344 <+205>: retq | End function. |

### CheckUserPermissonAccess Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 0x0000000000000f6a <+0>: push %rbp  0x0000000000000f6b <+1>: mov %rsp,%rbp  0x0000000000000f6e <+4>: push %rbx  0x0000000000000f6f <+5>: sub $0x48,%rsp  0x0000000000000f73 <+9>: mov %fs:0x28,%rax  0x0000000000000f7c <+18>: mov %rax,-0x18(%rbp)  0x0000000000000f80 <+22>: xor %eax,%eax | Setup function and allocate memory for local variables and setup overflow protection. |
| 0x0000000000000f82 <+24>: lea -0x45(%rbp),%rax  0x0000000000000f86 <+28>: mov %rax,%rdi  0x0000000000000f89 <+31>: callq 0xd20 <\_ZNSaIcEC1Ev@plt>  0x0000000000000f8e <+36>: lea -0x45(%rbp),%rdx  0x0000000000000f92 <+40>: lea -0x40(%rbp),%rax  0x0000000000000f96 <+44>: lea 0x588(%rip),%rsi # 0x1525  0x0000000000000f9d <+51>: mov %rax,%rdi  0x0000000000000fa0 <+54>: callq 0xce0 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEEC1EPKcRKS3\_@plt>  0x0000000000000fa5 <+59>: lea -0x45(%rbp),%rax  0x0000000000000fa9 <+63>: mov %rax,%rdi  0x0000000000000fac <+66>: callq 0xcb0 <\_ZNSaIcED1Ev@plt> | Print string to console. |
| 0x0000000000000fb1 <+71>: movl $0x0,-0x44(%rbp) | Set local variable -0x44(%rbp) to 0 |
| 0x0000000000000fb8 <+78>: lea 0x567(%rip),%rsi # 0x1526  0x0000000000000fbf <+85>: lea 0x2010da(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000000000000fc6 <+92>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Print string to console.  “Enter your username:” |
| 0x0000000000000fcb <+97>: lea 0x20130e(%rip),%rsi # 0x2022e0 <username>  0x0000000000000fd2 <+104>: lea 0x2011e7(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000000000000fd9 <+111>: callq 0xc40 <\_ZStrsIcSt11char\_traitsIcEERSt13basic\_istreamIT\_T0\_ES6\_PS3\_@plt> | Get user input for global variable username |
| 0x0000000000000fde <+116>: lea 0x558(%rip),%rsi # 0x153d  0x0000000000000fe5 <+123>: lea 0x2010b4(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000000000000fec <+130>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Print string to console.  “Enter your password:” |
| 0x0000000000000ff1 <+135>: lea -0x40(%rbp),%rax  0x0000000000000ff5 <+139>: mov %rax,%rsi  0x0000000000000ff8 <+142>: lea 0x2011c1(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  0x0000000000000fff <+149>: callq 0xcd0 <\_ZStrsIcSt11char\_traitsIcESaIcEERSt13basic\_istreamIT\_T0\_ES7\_RNSt7\_\_cxx1112basic\_stringIS4\_S5\_T1\_EE@plt> | Gets user input into local variable -0x40(%rbp) |
| 0x0000000000001004 <+154>: lea -0x40(%rbp),%rax  0x0000000000001008 <+158>: lea 0x545(%rip),%rsi # 0x1554  0x000000000000100f <+165>: mov %rax,%rdi  0x0000000000001012 <+168>: callq 0xc50 <\_ZNKSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEE7compareEPKc@plt> | Compares user input in -0x40(%rbp) to 0x545(%rip) |
| 0x0000000000001017 <+173>: mov %eax,-0x44(%rbp) | Sets local variable -0x44(%rbp) from above to return value of the function just called |
| 0x000000000000101a <+176>: cmpl $0x0,-0x44(%rbp)  0x000000000000101e <+180>: jne 0x1027 <\_Z25CheckUserPermissionAccessv+189> | if -0x44(%rbp) is not 0 jump to 189 |
| 0x0000000000001020 <+182>: mov $0x1,%ebx  0x0000000000001025 <+187>: jmp 0x102c <\_Z25CheckUserPermissionAccessv+194> | Set ebx register to 1 and jump to 194. This happens if -0x44(%rbp) is 0 |
| 0x0000000000001027 <+189>: mov $0x2,%ebx | Set ebx register to 2. This happens if -0x44(%rbp) is not 0 |
| 0x000000000000102c <+194>: lea -0x40(%rbp),%rax  0x0000000000001030 <+198>: mov %rax,%rdi  0x0000000000001033 <+201>: callq 0xc70 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEED1Ev@plt> |  |
| 0x0000000000001038 <+206>: mov %ebx,%eax | Tis moves ebx (1 or 2) into eax (the return register) |
| 0x000000000000103a <+208>: mov -0x18(%rbp),%rcx  0x000000000000103e <+212>: xor %fs:0x28,%rcx | Everything from here down is overflow protection handling, function clean up and returning eax. |
| 0x0000000000001047 <+221>: je 0x1084 <\_Z25CheckUserPermissionAccessv+282>  0x0000000000001049 <+223>: jmp 0x107f <\_Z25CheckUserPermissionAccessv+277>  0x000000000000104b <+225>: mov %rax,%rbx  0x000000000000104e <+228>: lea -0x45(%rbp),%rax  0x0000000000001052 <+232>: mov %rax,%rdi  0x0000000000001055 <+235>: callq 0xcb0 <\_ZNSaIcED1Ev@plt>  0x000000000000105a <+240>: mov %rbx,%rax  0x000000000000105d <+243>: mov %rax,%rdi  0x0000000000001060 <+246>: callq 0xd10 <\_Unwind\_Resume@plt>  0x0000000000001065 <+251>: mov %rax,%rbx  0x0000000000001068 <+254>: lea -0x40(%rbp),%rax  0x000000000000106c <+258>: mov %rax,%rdi  0x000000000000106f <+261>: callq 0xc70 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEED1Ev@plt>  0x0000000000001074 <+266>: mov %rbx,%rax  0x0000000000001077 <+269>: mov %rax,%rdi  0x000000000000107a <+272>: callq 0xd10 <\_Unwind\_Resume@plt>  0x000000000000107f <+277>: callq 0xcc0 <\_\_stack\_chk\_fail@plt>  0x0000000000001084 <+282>: add $0x48,%rsp  0x0000000000001088 <+286>: pop %rbx  0x0000000000001089 <+287>: pop %rbp  0x000000000000108a <+288>: retq |  |

### DisplayInfo Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 0x000000000000108b <+0>: push %rbp  0x000000000000108c <+1>: mov %rsp,%rbp | Set up the function |
| 0x000000000000108f <+4>: lea 0x4c2(%rip),%rsi # 0x1558  0x0000000000001096 <+11>: lea 0x201003(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x000000000000109d <+18>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x00000000000010a2 <+23>: mov %rax,%rdx  0x00000000000010a5 <+26>: mov 0x200f24(%rip),%rax # 0x201fd0  0x00000000000010ac <+33>: mov %rax,%rsi  0x00000000000010af <+36>: mov %rdx,%rdi  0x00000000000010b2 <+39>: callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Print string to console  “ Client's Name Service Selected (1 = Brokerage, 2 = Retirement)” << endl; |
| 0x00000000000010b7 <+44>: lea 0x4de(%rip),%rsi # 0x159c  0x00000000000010be <+51>: lea 0x200fdb(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x00000000000010c5 <+58>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x00000000000010ca <+63>: lea 0x200f3f(%rip),%rsi # 0x202010 <name1>  0x00000000000010d1 <+70>: mov %rax,%rdi  0x00000000000010d4 <+73>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x00000000000010d9 <+78>: lea 0x4c0(%rip),%rsi # 0x15a0  0x00000000000010e0 <+85>: mov %rax,%rdi  0x00000000000010e3 <+88>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x00000000000010e8 <+93>: mov %rax,%rdx  0x00000000000010eb <+96>: mov 0x200f6f(%rip),%eax # 0x202060 <num1>  0x00000000000010f1 <+102>: mov %eax,%esi  0x00000000000010f3 <+104>: mov %rdx,%rdi  0x00000000000010f6 <+107>: callq 0xd00 <\_ZNSolsEi@plt>  0x00000000000010fb <+112>: mov %rax,%rdx  0x00000000000010fe <+115>: mov 0x200ecb(%rip),%rax # 0x201fd0  0x0000000000001105 <+122>: mov %rax,%rsi  0x0000000000001108 <+125>: mov %rdx,%rdi  0x000000000000110b <+128>: callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Print string to console  “1. “ << name1 << “ selected option “ << num1 << endl; |
| 0x0000000000001110 <+133>: lea 0x49b(%rip),%rsi # 0x15b2  0x0000000000001117 <+140>: lea 0x200f82(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x000000000000111e <+147>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000000000001123 <+152>: lea 0x200ef6(%rip),%rsi # 0x202020 <name2>  0x000000000000112a <+159>: mov %rax,%rdi  0x000000000000112d <+162>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000000000001132 <+167>: lea 0x467(%rip),%rsi # 0x15a0  0x0000000000001139 <+174>: mov %rax,%rdi  0x000000000000113c <+177>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x0000000000001141 <+182>: mov %rax,%rdx  0x0000000000001144 <+185>: mov 0x200f1a(%rip),%eax # 0x202064 <num2>  0x000000000000114a <+191>: mov %eax,%esi  0x000000000000114c <+193>: mov %rdx,%rdi  0x000000000000114f <+196>: callq 0xd00 <\_ZNSolsEi@plt>  0x0000000000001154 <+201>: mov %rax,%rdx  0x0000000000001157 <+204>: mov 0x200e72(%rip),%rax # 0x201fd0  0x000000000000115e <+211>: mov %rax,%rsi  0x0000000000001161 <+214>: mov %rdx,%rdi  0x0000000000001164 <+217>: callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Print string to console  “2. “ << name2 << “ selected option “ << num2 << endl; |
| 0x0000000000001169 <+222>: lea 0x446(%rip),%rsi # 0x15b6  0x0000000000001170 <+229>: lea 0x200f29(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000000000001177 <+236>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x000000000000117c <+241>: lea 0x200ead(%rip),%rsi # 0x202030 <name3>  0x0000000000001183 <+248>: mov %rax,%rdi  0x0000000000001186 <+251>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x000000000000118b <+256>: lea 0x40e(%rip),%rsi # 0x15a0  0x0000000000001192 <+263>: mov %rax,%rdi  0x0000000000001195 <+266>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x000000000000119a <+271>: mov %rax,%rdx  0x000000000000119d <+274>: mov 0x200ec5(%rip),%eax # 0x202068 <num3>  0x00000000000011a3 <+280>: mov %eax,%esi  0x00000000000011a5 <+282>: mov %rdx,%rdi  0x00000000000011a8 <+285>: callq 0xd00 <\_ZNSolsEi@plt>  0x00000000000011ad <+290>: mov %rax,%rdx  0x00000000000011b0 <+293>: mov 0x200e19(%rip),%rax # 0x201fd0  0x00000000000011b7 <+300>: mov %rax,%rsi  0x00000000000011ba <+303>: mov %rdx,%rdi  0x00000000000011bd <+306>: callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Print string to console  “3. “ << name3 << “ selected option “ << num3 << endl; |
| 0x00000000000011c2 <+311>: lea 0x3f1(%rip),%rsi # 0x15ba  0x00000000000011c9 <+318>: lea 0x200ed0(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x00000000000011d0 <+325>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x00000000000011d5 <+330>: lea 0x200e64(%rip),%rsi # 0x202040 <name4>  0x00000000000011dc <+337>: mov %rax,%rdi  0x00000000000011df <+340>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x00000000000011e4 <+345>: lea 0x3b5(%rip),%rsi # 0x15a0  0x00000000000011eb <+352>: mov %rax,%rdi  0x00000000000011ee <+355>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x00000000000011f3 <+360>: mov %rax,%rdx  0x00000000000011f6 <+363>: mov 0x200e70(%rip),%eax # 0x20206c <num4>  0x00000000000011fc <+369>: mov %eax,%esi  0x00000000000011fe <+371>: mov %rdx,%rdi  0x0000000000001201 <+374>: callq 0xd00 <\_ZNSolsEi@plt>  0x0000000000001206 <+379>: mov %rax,%rdx  0x0000000000001209 <+382>: mov 0x200dc0(%rip),%rax # 0x201fd0  0x0000000000001210 <+389>: mov %rax,%rsi  0x0000000000001213 <+392>: mov %rdx,%rdi  0x0000000000001216 <+395>: callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Print string to console  “4. “ << name4 << “ selected option “ << num4 << endl; |
| 0x000000000000121b <+400>: lea 0x39c(%rip),%rsi # 0x15be  0x0000000000001222 <+407>: lea 0x200e77(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  0x0000000000001229 <+414>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x000000000000122e <+419>: lea 0x200e1b(%rip),%rsi # 0x202050 <name5>  0x0000000000001235 <+426>: mov %rax,%rdi  0x0000000000001238 <+429>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x000000000000123d <+434>: lea 0x35c(%rip),%rsi # 0x15a0  0x0000000000001244 <+441>: mov %rax,%rdi  0x0000000000001247 <+444>: callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  0x000000000000124c <+449>: mov %rax,%rdx  0x000000000000124f <+452>: mov 0x200e1b(%rip),%eax # 0x202070 <num5>  0x0000000000001255 <+458>: mov %eax,%esi  0x0000000000001257 <+460>: mov %rdx,%rdi  0x000000000000125a <+463>: callq 0xd00 <\_ZNSolsEi@plt>  0x000000000000125f <+468>: mov %rax,%rdx  0x0000000000001262 <+471>: mov 0x200d67(%rip),%rax # 0x201fd0  0x0000000000001269 <+478>: mov %rax,%rsi  0x000000000000126c <+481>: mov %rdx,%rdi  0x000000000000126f <+484>: callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Print string to console  “5. “ << name5 << “ selected option “ << num5 << endl; |
| 0x0000000000001274 <+489>: nop  0x0000000000001275 <+490>: pop %rbp  0x0000000000001276 <+491>: retq | Function ends and returns |
|  |  |