Project 1 - Proficiency Test

CS-410-R4890 Software Reserve Engineering

Raphael Coloma

April, 4 2024

# CS 410 Project One Proficiency Test Template

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

### “main” function”

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp | Stack Management for the main function |
| lea 0x5eb(%rip),%rsi # 0x1440  lea 0x201244(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 5eb(%rip) into register %rsi. Put memory address of 201244(%rip) into register %rdi. Print string “Hello! Welcome to our Inverstment Company\n” to screen. |
| callq 0xf6a <\_Z25CheckUserPermissionAccessv> | Call CheckUserPermissionAccess() function |
| mov %eax,0x201494(%rip) # 0x202300 <answer>  mov 0x20148e(%rip),%eax # 0x202300 <answer>  cmp $0x1,%eax  je 0xe8a <main+64> | Move value from register %eax to 0x201494 bytes above %rip. Move value from 0x20148e(%rip) to register %eax. Compare value 1 to the value in variable answer on register %eax. Jump to main+64 if values are equal. |
| lea 0x5f2(%rip),%rsi # 0x1470  lea 0x20121b(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 5f2(%rip) into register %rsi. Put memory address of 20121b(%rip) into register %rdi. Print string “Invalid Password. Please try again\n” to the screen. |
| mov 0x201470(%rip),%eax # 0x202300 <answer>  cmp $0x1,%eax  je 0xe97 <main+77> | Move value from 0x201470 bytes above %rip to register %eax. Compare value 1 to the value stored in variable answer on register %eax. Jump to main+77 if values are equal. |
| jmp 0xe61 <main+23> | Jump to main+23 (Call to CheckUserPermissionAccess) |
| lea 0x5f6(%rip),%rsi # 0x1494  lea 0x2011fb(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x5f6(%rip) into register %rsi. Put memory address of 0x2011fb(%rip) into register %rdi. Print string “What would you like to do?\n” to screen. |
| lea 0x5ff(%rip),%rsi # 0x14b0  lea 0x2011e8(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x5ff(%rip) into register %rsi. Put memory address of 0x2011e8(%rip) into register %rdi. Print string “DISPLAY the client list (enter 1)\n” to screen. |
| lea 0x614(%rip),%rsi # 0x14d8  lea 0x2011d5(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x614(%rip) into register %rsi. Put memory address of 0x2011d5(%rip) into register %rdi. Print string “CHANGE a client’s choice (enter 2)\n” to screen. |
| lea 0x625(%rip),%rsi # 0x14fc  lea 0x2011c2(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x625(%rip) into register %rsi. Put memory address of 0x2011c2(%rip) into register %rdi. Print string “Exit the program..\n” to screen. |
| lea 0x20140a(%rip),%rsi # 0x2022f4 <choice>  lea 0x2012cf(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0xc60 <\_ZNSirsERi@plt> | Put memory address of 0x20140a(%rip) into register %rsi. Put memory address of 0x2012cf(%rip) into register %rdi. Call cin and store user input to variable choice |
| lea 0x61d(%rip),%rsi # 0x151a  lea 0x20119c(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x61d(%rip) into register %rsi. Put memory address of 0x20119c(%rip) into register %rdi. Print string “You chose ” to screen. |
| mov %rax,%rdx  mov 0x2013e2(%rip),%eax # 0x2022f4 <choice>  mov %eax,%esi  mov %rdx,%rdi  callq 0xd00 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 0x2013e2(%rip) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print value of variable choice to screen |
| mov %rax,%rdx  mov 0x2010aa(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 0x2010aa(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character and Flush output buffer. |
| mov 0x2013bd(%rip),%eax # 0x2022f4 <choice>  cmp $0x1,%eax  jne 0xf43 <main+249> | Move value at 0x2013bd(%rip) to register %eax. Compare value 1 to value in variable choice on register %eax. Jump to main+265 if not equal. |
| callq 0x108b <\_Z11DisplayInfov> | Call function DisplayInfo() |
| jmp 0xf53 <main+265> | Jump to main+265 |
| mov 0x2013ab(%rip),%eax # 0x2022f4 <choice>  cmp $0x2,%eax  jne 0xf53 <main+265> | Move value at 0x2013ab(%rip) to register %eax. Compare value 2 to value in variable choice on register %eax. Jump to main+265 if not equal. |
| callq 0x1277 <\_Z20ChangeCustomerChoicev> | Call function ChangeCustomerChoice() |
| mov 0x20139b(%rip),%eax # 0x2022f4 <choice>  cmp $0x3,%eax  je 0xf63 <main+281> | Move value at 0x20139b(%rip) to register %eax. Compare value 3 to value in variable choice on register %eax. Jump to main+281 if equal. |
| jmpq 0xe97 <main+77> | While loop condition satisfied. Jump to main+77 |
| mov $0x0,%eax  pop %rbp  retq | Move value 0 to register %eax. Restore stack from register %rbp and return to main function. |

### ChangeCustomerChoice function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp | Stack Management for the ChangeCustomerChoice function |
| lea 0x346(%rip),%rsi # 0x15c8  lea 0x200e17(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x346(%rip) into register %rsi. Put memory address of 0x200e17(%rip) into register %rdi. Print string “Enter the number of the client that you wish to change\n” to the screen. |
| lea 0x201063(%rip),%rsi # 0x2022f8 <changechoice>  lea 0x200f24(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0xc60 <\_ZNSirsERi@plt> | Put memory address of 0x201063(%rip) into register %rsi. Put memory address of 0x200f24(%rip) into register %rdi. Call cin and store user input to global variable changechoice. |
| lea 0x358(%rip),%rsi # 0x1600  lea 0x200df1(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x358(%rip) into register %rsi. Put memory address of 0x200df1(%rip) into register %rdi. Print string “Please enter the client’s new service choice (1 = Brokerage, 2 = Retirement)\n” to the screen. |
| lea 0x201041(%rip),%rsi # 0x2022fc <newservice>  lea 0x200efe(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0xc60 <\_ZNSirsERi@plt> | Put memory address of 0x201041(%rip) into register %rsi. Put memory address of 0x200efe(%rip) into register %rdi. Call cin and store user input to global variable newservice. |
| mov 0x20102b(%rip),%eax # 0x2022f8 <changechoice>  cmp $0x1,%eax  jne 0x12e0 <\_Z20ChangeCustomerChoicev+105> | Move value from 0x20102b bytes above %rip to register %eax. Compare value 1 to variable changechoice on register %eax. Jump to \_Z20ChangeCustomerChoicev+105 if values not equal. |
| mov 0x201024(%rip),%eax # 0x2022fc <newservice>  mov %eax,0x200d82(%rip) # 0x202060 <num1>  jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Move value from 0x201024 bytes above %rip to register %eax. Move value from register %eax to 0x200d82 bytes above %rip num1 variable. Jump to \_Z20ChangeCustomerChoicev+203. |
| mov 0x201012(%rip),%eax # 0x2022f8 <changechoice>  cmp $0x2,%eax  jne 0x12f9 <\_Z20ChangeCustomerChoicev+130> | Move value from 0x201012 bytes above %rip to register %eax. Compare value 2 to variable changechoice on register %eax. Jump to \_Z20ChangeCustomerChoicev+130 if values not equal. |
| mov 0x20100b(%rip),%eax # 0x2022fc <newservice>  mov %eax,0x200d6d(%rip) # 0x202064 <num2>  jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Move value from 0x20100b bytes above %rip to register %eax. Move value from register %eax to 0x200d6d bytes above %rip to num2 variable. Jump to \_Z20ChangeCustomerChoicev+203. |
| mov 0x200ff9(%rip),%eax # 0x2022f8 <changechoice>  cmp $0x3,%eax  jne 0x1312 <\_Z20ChangeCustomerChoicev+155> | Move value from 0x200ff9 bytes above %rip to register %eax. Compare value 3 to variable changechoice on register %eax. Jump to \_Z20ChangeCustomerChoicev+155 if values not equal. |
| mov 0x200ff2(%rip),%eax # 0x2022fc <newservice>  mov %eax,0x200d58(%rip) # 0x202068 <num3>  jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Move value from 0x200ff2 bytes above %rip to register %eax. Move value from register %eax to 0x200d58 bytes above %rip to num3 variable. Jump to \_Z20ChangeCustomerChoicev+203. |
| mov 0x200fe0(%rip),%eax # 0x2022f8 <changechoice>  cmp $0x4,%eax  jne 0x132b <\_Z20ChangeCustomerChoicev+180> | Move value from 0x200fe0 bytes above %rip to register %eax. Compare value 4 to variable changechoice on register %eax. Jump to \_Z20ChangeCustomerChoicev+180 if values not equal. |
| mov 0x200fd9(%rip),%eax # 0x2022fc <newservice>  mov %eax,0x200d43(%rip) # 0x20206c <num4>  jmp 0x1342 <\_Z20ChangeCustomerChoicev+203> | Move value from 0x200fd9 bytes above %rip to register %eax. Move value from register %eax to 0x200d43 bytes above %rip to num4 variable. Jump to \_Z20ChangeCustomerChoicev+203. |
| mov 0x200fc7(%rip),%eax # 0x2022f8 <changechoice>  cmp $0x5,%eax  jne 0x1342 <\_Z20ChangeCustomerChoicev+203> | Move value from 0x200fc7 bytes above %rip to register %eax. Compare value 5 to variable changechoice on register %eax. Jump to \_Z20ChangeCustomerChoicev+203 if values not equal. |
| mov 0x200fc0(%rip),%eax # 0x2022fc <newservice>  mov %eax,0x200d2e(%rip) # 0x202070 <num5> | Move value from 0x200fc0 bytes above %rip to register %eax. Move value from register %eax to 0x200d2e bytes above %rip to num5 variable. Jump to \_Z20ChangeCustomerChoicev+203. |
| nop | No operation |
| pop %rbp  retq | Restore stack from register %rbp and return to main function. |

### CheckUserPermissionAccess Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp  push %rbx | Stack Management for CheckUserPermissionAccess function. |
| sub $0x48,%rsp | Reserve space on stack for variable declarations. |
| mov %fs:0x28,%rax  mov %rax,-0x18(%rbp)  xor %eax,%eax | Store sentinel stack-guard value. Exclusive or register %eax by itself to set it to 0. |
| lea -0x45(%rbp),%rax  mov %rax,%rdi  callq 0xd20 <\_ZNSaIcEC1Ev@plt> | Put memory address of 45(%rbp) into register %rax. Move value from register %rax into register %rdi. Call unknown string function. |
| lea -0x45(%rbp),%rdx  lea -0x40(%rbp),%rax  lea 0x588(%rip),%rsi # 0x1525  mov %rax,%rdi  callq 0xce0 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEEC1EPKcRKS3\_@plt> | Put memory location of -0x45(%rbp) in register %rdx. Put memory location of -0x40(%rbp) in register %rax. Put memory location of 0x588(%rbp) in register %rsi. Move value in register %rax to register %rdi. Call assign password variable to null string |
| lea -0x45(%rbp),%rax  mov %rax,%rdi  callq 0xcb0 <\_ZNSaIcED1Ev@plt> | Put memory address of 45(%rbp) into register %rax. Move value from register %rax into register %rdi Call unknown string function. |
| movl $0x0,-0x44(%rbp)  lea 0x567(%rip),%rsi # 0x1526  lea 0x2010da(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Move long value 0 to -0x44 bytes above register %rbp. Put memory address of 0x567(%rip) into register %rsi. Put memory address of 2010da(%rip) into register %rdi. Print string “Enter your username:\n” to screen. |
| lea 0x20130e(%rip),%rsi # 0x2022e0 <username>  lea 0x2011e7(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0xc40 <\_ZStrsIcSt11char\_traitsIcEERSt13basic\_istreamIT\_T0\_ES6\_PS3\_@plt> | Put memory address of 45(%rbp) into register %rdx. Put memory address of 40(%rbp) into register %rax. Put memory address of 588(%rip) into register %rsi. Move value from register %rax to register %rdi. Call cin function for and store in variable username. |
| lea 0x558(%rip),%rsi # 0x153d  lea 0x2010b4(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 0x558(%rip) into register %rsi. Put memory address of 0x2010b4(%rip) into register %rdi. Print string “Enter your password: \n” to the screen. |
| lea -0x40(%rbp),%rax  mov %rax,%rsi  lea 0x2011c1(%rip),%rdi # 0x2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 0xcd0 <\_ZStrsIcSt11char\_traitsIcESaIcEERSt13basic\_istreamIT\_T0\_ES7\_RNSt7\_\_cxx1112basic\_stringIS4\_S5\_T1\_EE@plt> | Put memory address of 40(%rbp) into register %rax.  Move value from register %rax to register %rdi. Put memory address of 0x2011c1(%rip) into register %rdi. Call cin function for user input. |
| lea -0x40(%rbp),%rax  lea 0x545(%rip),%rsi # 0x1554  mov %rax,%rdi  callq 0xc50 <\_ZNKSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEE7compareEPKc@plt> | Put memory address of 40(%rbp) into register %rax.  Put memory address of 0x545(%rip) into register %rsi. Move value from register %rax to register %rdi. Call function to compare strings. |
| mov %eax,-0x44(%rbp)  cmpl $0x0,-0x44(%rbp)  jne 0x1027 <\_Z25CheckUserPermissionAccessv+189> | Move value from register %eax to 0x44 bytes above register %rbp. Compare long value 0 to the value at 0x44 bytes above register %rbp. Jump to \_Z25CheckUserPermissionAccessv+189 if not equal. |
| mov $0x1,%ebx  jmp 0x102c <\_Z25CheckUserPermissionAccessv+194> | Move value 1 to register %ebx. Jump to \_Z25CheckUserPermissionAccessv+194 |
| mov $0x2,%ebx  lea -0x40(%rbp),%rax  mov %rax,%rdi  callq 0xc70 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEED1Ev@plt> | Move value 2 to register %ebx. Put memory location of -0x40(%rbp) in register %rax. Move value in register %rax to register %rdi. Call unknown string function. |
| mov %ebx,%eax  mov -0x18(%rbp),%rcx  xor %fs:0x28,%rcx  je 0x1084 <\_Z25CheckUserPermissionAccessv+282> | Move the value from register %ebx to register %eax. Move the value at-0x18(%rbp) to register %rcx. Exclusive or values at sentinel stack-guard value %fs:0x28 and register %rcx. Jump to CheckUserPermissionAccess+282 if values are equal. |
| jmp 0x107f <\_Z25CheckUserPermissionAccessv+277> | Jump to CheckUserPermissionAccess+277 |
| mov %rax,%rbx  lea -0x45(%rbp),%rax  mov %rax,%rdi  callq 0xcb0 <\_ZNSaIcED1Ev@plt> | Move value 2 to register %ebx. Put memory location of -0x40(%rbp) in register %rax. Move value in register %rax to register %rdi. Call unknown string function. |
| mov %rbx,%rax  mov %rax,%rdi  callq 0xd10 <\_Unwind\_Resume@plt> | Move the value from register %rbx to register %rax. Move the value from register %rax to register %rdi. Call error handling method Unwind\_Resume |
| mov %rax,%rbx  lea -0x40(%rbp),%rax  mov %rax,%rdi  callq 0xc70 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEED1Ev@plt> | Move the value at register %rax to register %rbx. Put memory location of -0x40(%rbp) in register %rax. Move value in register %rax to register %rdi. Call unknown string function. |
| mov %rbx,%rax  mov %rax,%rdi  callq 0xd10 <\_Unwind\_Resume@plt> | Move the value from register %rbx to register %rax. Move the value from register %rax to register %rdi. Call error handling method Unwind\_Resume |
| callq 0xcc0 <\_\_stack\_chk\_fail@plt> | Call stack\_chk\_fail function - stack protect guard variable has been modified |
| add $0x48,%rsp  pop %rbx  pop %rbp  retq | Restore stack from register %rbp and %rbx. Return value to main function. |

### DisplayInfo Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp | Stack Management for DisplayInfo function |
| lea 0x4c2(%rip),%rsi # 0x1558  lea 0x201003(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 4c2(%rbp) into register %rsi. Put memory address of 201003(%rip) into register %rdi. Print string “ Client’s Name Service Selected (1 = Brokerage, 2 = Retirement)” to screen. |
| mov %rax,%rdx  mov 0x200f24(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 200f24(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character and Flush output buffer. |
| lea 0x4de(%rip),%rsi # 0x159c  lea 0x200fdb(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 4de(%rip) into register %rsi. Put memory address of 200fdb(%rip) into register %rdi. Print string “1. ” to screen. |
| lea 0x200f3f(%rip),%rsi # 0x202010 <name1>  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 200f3f(%rip) into register %rsi. Move value from register %rax to register %rdi. Print value from global variable name1 to screen. |
| lea 0x4c0(%rip),%rsi # 0x15a0  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 4c0(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ selected option ” to screen. |
| mov %rax,%rdx  mov 0x200f6f(%rip),%eax # 0x202060 <num1>  mov %eax,%esi  mov %rdx,%rdi  callq 0xd00 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 200f6f(%rip) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print value of global variable num1 to screen |
| mov %rax,%rdx  mov 0x200ecb(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 200ecb(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character and Flush output buffer. |
| lea 0x49b(%rip),%rsi # 0x15b2  lea 0x200f82(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 49b(%rip) into register %rsi. Put memory address of 200f82(%rip) into register %rdi. Print string “2. ” to screen. |
| lea 0x200ef6(%rip),%rsi # 0x202020 <name2>  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 200ef6(%rip) into register %rsi. Move value from register %rax to register %rdi. Print value from global variable name2 to screen. |
| lea 0x467(%rip),%rsi # 0x15a0  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 467(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ selected option ” to screen. |
| mov %rax,%rdx  mov 0x200f1a(%rip),%eax # 0x202064 <num2>  mov %eax,%esi  mov %rdx,%rdi  callq 0xd00 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 200f1a(%rip) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print value of global variable num2 to screen |
| mov %rax,%rdx  mov 0x200e72(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 200e72(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character and Flush output buffer. |
| lea 0x446(%rip),%rsi # 0x15b6  lea 0x200f29(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 446(%rip) into register %rsi. Put memory address of 200f29(%rip) into register %rdi. Print string “3. ” to screen. |
| lea 0x200ead(%rip),%rsi # 0x202030 <name3>  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 200ead(%rip) into register %rsi. Move value from register %rax to register %rdi. Print value from global variable name3 to screen. |
| lea 0x40e(%rip),%rsi # 0x15a0  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 140e(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ selected option ” to screen. |
| mov %rax,%rdx  mov 0x200ec5(%rip),%eax # 0x202068 <num3>  mov %eax,%esi  mov %rdx,%rdi  callq 0xd00 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 200ec5(%rip) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print value of global variable num3 to screen |
| mov %rax,%rdx  mov 0x200e19(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 200e19(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character and Flush output buffer. |
| lea 0x3f1(%rip),%rsi # 0x15ba  lea 0x200ed0(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 3f1(%rip) into register %rsi. Put memory address of 200ed0(%rip) into register %rdi. Print string “4. ” to screen. |
| lea 0x200e64(%rip),%rsi # 0x202040 <name4>  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 200e64(%rip) into register %rsi. Move value from register %rax to register %rdi. Print value from global variable name4 to screen. |
| lea 0x3b5(%rip),%rsi # 0x15a0  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 3b5(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ selected option ” to screen. |
| mov %rax,%rdx  mov 0x200e70(%rip),%eax # 0x20206c <num4>  mov %eax,%esi  mov %rdx,%rdi  callq 0xd00 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 200e70(%rip) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print value of global variable num4 to screen |
| mov %rax,%rdx  mov 0x200dc0(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 200dc0(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character and Flush output buffer. |
| lea 0x39c(%rip),%rsi # 0x15be  lea 0x200e77(%rip),%rdi # 0x2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 039c(%rip) into register %rsi. Put memory address of 200e77(%rip) into register %rdi. Print string “5. ” to screen. |
| lea 0x200e1b(%rip),%rsi # 0x202050 <name5>  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 200e1b(%rip) into register %rsi. Move value from register %rax to register %rdi. Print value from global variable name5 to screen. |
| lea 0x35c(%rip),%rsi # 0x15a0  mov %rax,%rdi  callq 0xc90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Put memory address of 35c(%rip) into register %rsi. Move value from register %rax to register %rdi. Print string “ selected option ” to screen. |
| mov %rax,%rdx  mov 0x200e1b(%rip),%eax # 0x202070 <num5>  mov %eax,%esi  mov %rdx,%rdi  callq 0xd00 <\_ZNSolsEi@plt> | Move value from register %rax to register %rdx. Move value from 200e1b(%rip) to register %eax. Move value from register %eax to register %esi. Move value from register %rdx to register %rdi. Print value of global variable num5 to screen |
| mov %rax,%rdx  mov 0x200d67(%rip),%rax # 0x201fd0  mov %rax,%rsi  mov %rdx,%rdi  callq 0xca0 <\_ZNSolsEPFRSoS\_E@plt> | Move value from register %rax to register %rdx. Move value from 200d67(%rip) to register %rax. Move value from register %rax to register %rsi. Move value from register %rdx to register %rdi. Print newline character and Flush output buffer. |
| nop | No operation |
| pop %rbp  retq | Restore stack from register %rbp and return to main function. |