# 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 | Push the current base pointer (rbp) onto the stack. |
| mov %rsp,%rbp | Set the stack pointer (rsp) to the base pointer (rbp). |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq 17 <main+0x17> | Call the function at offset 17. |
| callq 1c <main+0x1c> | Call the function at offset 1c. |
| mov %eax,0x0(%rip) | Store the value in %eax into the memory location pointed to by the current instruction pointer address. |
| mov 0x0(%rip),%eax | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x1,%eax | Compare the value in %eax to 1. |
| je 40 <main+0x40> | Jump to offset 40 if the previous comparison was equal. |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq 40 <main+0x40> | Call the function at offset 40. |
| mov 0x0(%rip),%eax | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x1,%eax | Compare the value in %eax to 1. |
| je 4d <main+0x4d> | Jump to offset 4d if the previous comparison was equal. |
| jmp 17 <main+0x17> | Jump unconditionally to offset 17. |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq 60 <main+0x60> | Call the function at offset 60. |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq 73 <main+0x73> | Call the function at offset 73. |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq 86 <main+0x86> | Call the function at offset 86. |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq 99 <main+0x99> | Call the function at offset 99. |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq ac <main+0xac> | Call the function at offset ac. |
| lea 0x0(%rip),%rsi | Load the current instruction pointer address into %rsi. |
| lea 0x0(%rip),%rdi | Load the current instruction pointer address into %rdi. |
| callq bf <main+0xbf> | Call the function at offset bf. |
| mov %rax,%rdx | Move the value in %rax to %rdx. |
| mov 0x0(%rip),%eax | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| mov %eax,%esi | Move the value in %eax to %esi. |
| mov %rdx,%rdi | Move the value in %rdx to %rdi. |
| callq d2 <main+0xd2> | Call the function at offset d2. |
| mov %rax,%rdx | Move the value in %rax to %rdx. |
| mov 0x0(%rip),%rax | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| mov %rax,%rsi | Move the value in %rax to %rsi. |
| mov %rdx,%rdi | Move the value in %rdx to %rdi. |
| callq e7 <main+0xe7> | Call the function at offset e7. |
| mov 0x0(%rip),%eax | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x1,%eax | Compare the value in %eax to 1. |
| jne f9 <main+0xf9> | Jump to offset f9 if the previous comparison was not equal. |
| callq f7 <main+0xf7> | Call the function at offset f7. |
| jmp 109 <main+0x109> | Jump unconditionally to offset 109. |
| mov 0x0(%rip),%eax | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x2,%eax | Compare the value in %eax to 2. |
| jne 109 <main+0x109> | Jump to offset 109 if the previous comparison was not equal. |
| callq 109 <main+0x109> | Call the function at offset 109. |
| mov 0x0(%rip),%eax | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x3,%eax | Compare the value in %eax to 3. |
| je 119 <main+0x119> | Jump to offset 119 if the previous comparison was equal. |
| jmpq 4d <main+0x4d> | Jump unconditionally to offset 4d. |
| mov $0x0,%eax | Set %eax to 0. |
| pop %rbp | Restore the base pointer from the stack. |
| retq | Return from the function. |

### ChangeCustomerChoice function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |

|  |  |
| --- | --- |
| push %rbp | Push the current base pointer (rbp) onto the stack. |
| mov %rsp,%rbp | Set the base pointer (rbp) to the value of the stack pointer (rsp). |
| lea 0x0(%rip),%rsi # 438 <\_Z20ChangeCustomerChoicev+0xb> | Load the address pointed to by the current instruction pointer into %rsi. |
| lea 0x0(%rip),%rdi # 43f <\_Z20ChangeCustomerChoicev+0x12> | Load the address pointed to by the current instruction pointer into %rdi. |
| callq 444 <\_Z20ChangeCustomerChoicev+0x17> | Call the function at offset 444. |
| lea 0x0(%rip),%rsi # 44b <\_Z20ChangeCustomerChoicev+0x1e> | Load the address pointed to by the current instruction pointer into %rsi. |
| lea 0x0(%rip),%rdi # 452 <\_Z20ChangeCustomerChoicev+0x25> | Load the address pointed to by the current instruction pointer into %rdi. |
| callq 457 <\_Z20ChangeCustomerChoicev+0x2a> | Call the function at offset 457. |
| lea 0x0(%rip),%rsi # 45e <\_Z20ChangeCustomerChoicev+0x31> | Load the address pointed to by the current instruction pointer into %rsi. |
| lea 0x0(%rip),%rdi # 465 <\_Z20ChangeCustomerChoicev+0x38> | Load the address pointed to by the current instruction pointer into %rdi. |
| callq 46a <\_Z20ChangeCustomerChoicev+0x3d> | Call the function at offset 46a. |
| lea 0x0(%rip),%rsi # 471 <\_Z20ChangeCustomerChoicev+0x44> | Load the address pointed to by the current instruction pointer into %rsi. |
| lea 0x0(%rip),%rdi # 478 <\_Z20ChangeCustomerChoicev+0x4b> | Load the address pointed to by the current instruction pointer into %rdi. |
| callq 47d <\_Z20ChangeCustomerChoicev+0x50> | Call the function at offset 47d. |
| mov 0x0(%rip),%eax # 483 <\_Z20ChangeCustomerChoicev+0x56> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x1,%eax | Compare the value in %eax to 1. |
| jne 496 <\_Z20ChangeCustomerChoicev+0x69> | Jump to offset 496 if the previous comparison was not equal. |
| mov 0x0(%rip),%eax # 48e <\_Z20ChangeCustomerChoicev+0x61> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| mov %eax,0x0(%rip) # 494 <\_Z20ChangeCustomerChoicev+0x67> | Store the value from %eax to the memory location pointed to by the current instruction pointer address. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the offset 4f8. |
| mov 0x0(%rip),%eax # 49c <\_Z20ChangeCustomerChoicev+0x6f> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x2,%eax | Compare the value in %eax to 2. |
| jne 4af <\_Z20ChangeCustomerChoicev+0x82> | Jump to offset 4af if the previous comparison was not equal. |
| mov 0x0(%rip),%eax # 4a7 <\_Z20ChangeCustomerChoicev+0x7a> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| mov %eax,0x0(%rip) # 4ad <\_Z20ChangeCustomerChoicev+0x80> | Store the value from %eax to the memory location pointed to by the current instruction pointer address. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the offset 4f8. |
| mov 0x0(%rip),%eax # 4b5 <\_Z20ChangeCustomerChoicev+0x88> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x3,%eax | Compare the value in %eax to 3. |
| jne 4c8 <\_Z20ChangeCustomerChoicev+0x9b> | Jump to offset 4c8 if the previous comparison was not equal. |
| mov 0x0(%rip),%eax # 4c0 <\_Z20ChangeCustomerChoicev+0x93> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| mov %eax,0x0(%rip) # 4c6 <\_Z20ChangeCustomerChoicev+0x99> | Store the value from %eax to the memory location pointed to by the current instruction pointer address. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the offset 4f8. |
| mov 0x0(%rip),%eax # 4ce <\_Z20ChangeCustomerChoicev+0xa1> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x4,%eax | Compare the value in %eax to 4. |
| jne 4e1 <\_Z20ChangeCustomerChoicev+0xb4> | Jump to offset 4e1 if the previous comparison was not equal. |
| mov 0x0(%rip),%eax # 4d9 <\_Z20ChangeCustomerChoicev+0xac> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| mov %eax,0x0(%rip) # 4df <\_Z20ChangeCustomerChoicev+0xb2> | Store the value from %eax to the memory location pointed to by the current instruction pointer address. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the offset 4f8. |
| mov 0x0(%rip),%eax # 4e7 <\_Z20ChangeCustomerChoicev+0xba> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| cmp $0x5,%eax | Compare the value in %eax to 5. |
| jne 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Jump to offset 4f8 if the previous comparison was not equal. |
| mov 0x0(%rip),%eax # 4f2 <\_Z20ChangeCustomerChoicev+0xc5> | Load the value from the memory location pointed to by the current instruction pointer address into %eax. |
| mov %eax,0x0(%rip) # 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Store the value from %eax to the memory location pointed to by the current instruction pointer address. |
| nop | No operation. This instruction does nothing and is typically used for padding. |
| pop %rbp | Pop the top value from the stack into the base pointer (rbp). |
| retq | Return from the function. |

### CheckUserPermissonAccess Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |

|  |  |
| --- | --- |
| push %rbp | Pushes the value of the base pointer (rbp) onto the stack. |
| mov %rsp,%rbp | Copies the current value of the stack pointer (rsp) to the base pointer (rbp). |
| lea 0x0(%rip),%rsi # 438 <\_Z20ChangeCustomerChoicev+0xb> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 43f <\_Z20ChangeCustomerChoicev+0x12> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 444 <\_Z20ChangeCustomerChoicev+0x17> | Calls the function at address 444. |
| lea 0x0(%rip),%rsi # 44b <\_Z20ChangeCustomerChoicev+0x1e> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 452 <\_Z20ChangeCustomerChoicev+0x25> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 457 <\_Z20ChangeCustomerChoicev+0x2a> | Calls the function at address 457. |
| lea 0x0(%rip),%rsi # 45e <\_Z20ChangeCustomerChoicev+0x31> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 465 <\_Z20ChangeCustomerChoicev+0x38> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 46a <\_Z20ChangeCustomerChoicev+0x3d> | Calls the function at address 46a. |
| lea 0x0(%rip),%rsi # 471 <\_Z20ChangeCustomerChoicev+0x44> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 478 <\_Z20ChangeCustomerChoicev+0x4b> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 47d <\_Z20ChangeCustomerChoicev+0x50> | Calls the function at address 47d. |
| mov 0x0(%rip),%eax # 483 <\_Z20ChangeCustomerChoicev+0x56> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| cmp $0x1,%eax | Compares the value in %eax to 1. |
| jne 496 <\_Z20ChangeCustomerChoicev+0x69> | If the previous comparison is not equal, jump to the instruction at address 496. |
| mov 0x0(%rip),%eax # 48e <\_Z20ChangeCustomerChoicev+0x61> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,0x0(%rip) # 494 <\_Z20ChangeCustomerChoicev+0x67> | Stores the value in %eax to the address specified by the current instruction pointer. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the instruction at address 4f8. |
| mov 0x0(%rip),%eax # 49c <\_Z20ChangeCustomerChoicev+0x6f> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| cmp $0x2,%eax | Compares the value in %eax to 2. |
| jne 4af <\_Z20ChangeCustomerChoicev+0x82> | If the previous comparison is not equal, jump to the instruction at address 4af. |
| mov 0x0(%rip),%eax # 4a7 <\_Z20ChangeCustomerChoicev+0x7a> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,0x0(%rip) # 4ad <\_Z20ChangeCustomerChoicev+0x80> | Stores the value in %eax to the address specified by the current instruction pointer. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the instruction at address 4f8. |
| mov 0x0(%rip),%eax # 4b5 <\_Z20ChangeCustomerChoicev+0x88> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| cmp $0x3,%eax | Compares the value in %eax to 3. |
| jne 4c8 <\_Z20ChangeCustomerChoicev+0x9b> | If the previous comparison is not equal, jump to the instruction at address 4c8. |
| mov 0x0(%rip),%eax # 4c0 <\_Z20ChangeCustomerChoicev+0x93> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,0x0(%rip) # 4c6 <\_Z20ChangeCustomerChoicev+0x99> | Stores the value in %eax to the address specified by the current instruction pointer. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the instruction at address 4f8. |
| mov 0x0(%rip),%eax # 4ce <\_Z20ChangeCustomerChoicev+0xa1> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| cmp $0x4,%eax | Compares the value in %eax to 4. |
| jne 4e1 <\_Z20ChangeCustomerChoicev+0xb4> | If the previous comparison is not equal, jump to the instruction at address 4e1. |
| mov 0x0(%rip),%eax # 4d9 <\_Z20ChangeCustomerChoicev+0xac> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,0x0(%rip) # 4df <\_Z20ChangeCustomerChoicev+0xb2> | Stores the value in %eax to the address specified by the current instruction pointer. |
| jmp 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Unconditionally jump to the instruction at address 4f8. |
| mov 0x0(%rip),%eax # 4e7 <\_Z20ChangeCustomerChoicev+0xba> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| cmp $0x5,%eax | Compares the value in %eax to 5. |
| jne 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | If the previous comparison is not equal, jump to the instruction at address 4f8. |
| mov 0x0(%rip),%eax # 4f2 <\_Z20ChangeCustomerChoicev+0xc5> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,0x0(%rip) # 4f8 <\_Z20ChangeCustomerChoicev+0xcb> | Stores the value in %eax to the address specified by the current instruction pointer. |
| nop | No operation is performed. This is often used for alignment purposes. |
| pop %rbp | Pops the top value from the stack into the base pointer (rbp). |
| retq | Returns from the current function. |

### DisplayInfo Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |

|  |  |
| --- | --- |
| push %rbp | Pushes the base pointer (rbp) onto the stack. This saves the current base pointer. |
| mov %rsp,%rbp | Moves the stack pointer (rsp) to the base pointer (rbp). This sets up the new stack frame. |
| lea 0x0(%rip),%rsi # 24c <\_Z11DisplayInfov+0xb> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 253 <\_Z11DisplayInfov+0x12> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 258 <\_Z11DisplayInfov+0x17> | Calls the function at address 258. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%rax # 262 <\_Z11DisplayInfov+0x21> | Loads the value at the address specified by the current instruction pointer into the %rax register. |
| mov %rax,%rsi | Moves the value in %rax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 26d <\_Z11DisplayInfov+0x2c> | Calls the function at address 26d. |
| lea 0x0(%rip),%rsi # 274 <\_Z11DisplayInfov+0x33> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 27b <\_Z11DisplayInfov+0x3a> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 280 <\_Z11DisplayInfov+0x3f> | Calls the function at address 280. |
| lea 0x0(%rip),%rsi # 287 <\_Z11DisplayInfov+0x46> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 28f <\_Z11DisplayInfov+0x4e> | Calls the function at address 28f. |
| lea 0x0(%rip),%rsi # 296 <\_Z11DisplayInfov+0x55> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 29e <\_Z11DisplayInfov+0x5d> | Calls the function at address 29e. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%eax # 2a7 <\_Z11DisplayInfov+0x66> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,%esi | Moves the value in %eax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 2b1 <\_Z11DisplayInfov+0x70> | Calls the function at address 2b1. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%rax # 2bb <\_Z11DisplayInfov+0x7a> | Loads the value at the address specified by the current instruction pointer into the %rax register. |
| mov %rax,%rsi | Moves the value in %rax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 2c6 <\_Z11DisplayInfov+0x85> | Calls the function at address 2c6. |
| lea 0x0(%rip),%rsi # 2cd <\_Z11DisplayInfov+0x8c> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 2d4 <\_Z11DisplayInfov+0x93> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 2d9 <\_Z11DisplayInfov+0x98> | Calls the function at address 2d9. |
| lea 0x0(%rip),%rsi # 2e0 <\_Z11DisplayInfov+0x9f> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 2e8 <\_Z11DisplayInfov+0xa7> | Calls the function at address 2e8. |
| lea 0x0(%rip),%rsi # 2ef <\_Z11DisplayInfov+0xae> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 2f7 <\_Z11DisplayInfov+0xb6> | Calls the function at address 2f7. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%eax # 300 <\_Z11DisplayInfov+0xbf> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,%esi | Moves the value in %eax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 30a <\_Z11DisplayInfov+0xc9> | Calls the function at address 30a. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%rax # 314 <\_Z11DisplayInfov+0xd3> | Loads the value at the address specified by the current instruction pointer into the %rax register. |
| mov %rax,%rsi | Moves the value in %rax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 31f <\_Z11DisplayInfov+0xde> | Calls the function at address 31f. |
| lea 0x0(%rip),%rsi # 326 <\_Z11DisplayInfov+0xe5> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 32d <\_Z11DisplayInfov+0xec> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 332 <\_Z11DisplayInfov+0xf1> | Calls the function at address 332. |
| lea 0x0(%rip),%rsi # 339 <\_Z11DisplayInfov+0xf8> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 341 <\_Z11DisplayInfov+0x100> | Calls the function at address 341. |
| lea 0x0(%rip),%rsi # 348 <\_Z11DisplayInfov+0x107> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 350 <\_Z11DisplayInfov+0x10f> | Calls the function at address 350. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%eax # 359 <\_Z11DisplayInfov+0x118> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,%esi | Moves the value in %eax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 363 <\_Z11DisplayInfov+0x122> | Calls the function at address 363. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%rax # 36d <\_Z11DisplayInfov+0x12c> | Loads the value at the address specified by the current instruction pointer into the %rax register. |
| mov %rax,%rsi | Moves the value in %rax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 378 <\_Z11DisplayInfov+0x137> | Calls the function at address 378. |
| lea 0x0(%rip),%rsi # 37f <\_Z11DisplayInfov+0x13e> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 386 <\_Z11DisplayInfov+0x145> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 38b <\_Z11DisplayInfov+0x14a> | Calls the function at address 38b. |
| lea 0x0(%rip),%rsi # 392 <\_Z11DisplayInfov+0x151> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 39a <\_Z11DisplayInfov+0x159> | Calls the function at address 39a. |
| lea 0x0(%rip),%rsi # 3a1 <\_Z11DisplayInfov+0x160> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 3a9 <\_Z11DisplayInfov+0x168> | Calls the function at address 3a9. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%eax # 3b2 <\_Z11DisplayInfov+0x171> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,%esi | Moves the value in %eax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 3bc <\_Z11DisplayInfov+0x17b> | Calls the function at address 3bc. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%rax # 3c6 <\_Z11DisplayInfov+0x185> | Loads the value at the address specified by the current instruction pointer into the %rax register. |
| mov %rax,%rsi | Moves the value in %rax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 3d1 <\_Z11DisplayInfov+0x190> | Calls the function at address 3d1. |
| lea 0x0(%rip),%rsi # 3d8 <\_Z11DisplayInfov+0x197> | Loads the effective address of the current instruction pointer into the %rsi register. |
| lea 0x0(%rip),%rdi # 3df <\_Z11DisplayInfov+0x19e> | Loads the effective address of the current instruction pointer into the %rdi register. |
| callq 3e4 <\_Z11DisplayInfov+0x1a3> | Calls the function at address 3e4. |
| lea 0x0(%rip),%rsi # 3eb <\_Z11DisplayInfov+0x1aa> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 3f3 <\_Z11DisplayInfov+0x1b2> | Calls the function at address 3f3. |
| lea 0x0(%rip),%rsi # 3fa <\_Z11DisplayInfov+0x1b9> | Loads the effective address of the current instruction pointer into the %rsi register. |
| mov %rax,%rdi | Moves the value in %rax to %rdi. |
| callq 402 <\_Z11DisplayInfov+0x1c1> | Calls the function at address 402. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%eax # 40b <\_Z11DisplayInfov+0x1ca> | Loads the value at the address specified by the current instruction pointer into the %eax register. |
| mov %eax,%esi | Moves the value in %eax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 415 <\_Z11DisplayInfov+0x1d4> | Calls the function at address 415. |
| mov %rax,%rdx | Moves the value in %rax to %rdx. |
| mov 0x0(%rip),%rax # 41f <\_Z11DisplayInfov+0x1de> | Loads the value at the address specified by the current instruction pointer into the %rax register. |
| mov %rax,%rsi | Moves the value in %rax to %rsi. |
| mov %rdx,%rdi | Moves the value in %rdx to %rdi. |
| callq 42a <\_Z11DisplayInfov+0x1e9> | Calls the function at address 42a. |
| nop | No operation. It does nothing and moves to the next instruction. |
| pop %rbp | Pops the top of the stack into the base pointer (rbp). This restores the old base pointer. |
| retq | Returns from the current function. |