# CS 410 Project One Proficiency Test

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

### “main” function”

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 0: push %rbp | Saves the current base pointer to preserve the previous stack frame. |
| 1: mov %rsp, %rbp | Creates a new stack frame by moving the stack pointer to the base pointer. |
| 4: lea 0x0(%rip), %rsi | Loads an address (likely a string or data) into the %rsi register for a function argument. |
| b: lea 0x0(%rip), %rdi | Loads an address into the %rdi register as another function argument. |
| 12: call 17 | Calls the function located at address 0x17. |
| 17: call 1c | Calls the function at address 0x1c. |
| 1c: mov %eax, 0x0(%rip) | Stores the value in %eax to a memory location for later use. |
| 22: mov 0x0(%rip), %eax | Loads a value from memory into the %eax register. |
| 28: cmp $0x1, %eax | Compares the value in %eax to 1. |
| 2b: je 40 | Jumps to address 0x40 if the value equals 1. |
| 2d: lea 0x0(%rip), %rsi | Loads a new value into %rsi as a function argument. |
|  |  |
| 34: lea 0x0(%rip), %rdi | Loads a new value into %rdi as a second argument. |
| 3b: call 40 | Calls the function at address 0x40. |
| 40: mov 0x0(%rip), %eax | Loads a value from memory into %eax. |
| 46: cmp $0x1, %eax | Compares the new value in %eax to 1. |
| 49: je 4d | Jumps to address 0x4d if comparison is true. |
| 4b: jmp 17 | Unconditionally jumps back to 0x17, creating a loop. |
| 4d: lea 0x0(%rip), %rsi | Loads an address into %rsi for another function call. |
| 54: lea 0x0(%rip), %rdi | Loads another address into %rdi as an argument. |
|  |  |
| 5b: call 60 | Calls the function at address 0x60. |
| 60: lea 0x0(%rip), %rsi | Loads another address into %rsi. |
| 67: lea 0x0(%rip), %rdi | Loads another value into %rdi. |
| 6e: call 73 | Calls the function at address 0x73. |
| 73: lea 0x0(%rip), %rsi | Prepares next function argument in %rsi. |
| 7a: lea 0x0(%rip), %rdi | Prepares next argument in %rdi. |
| 81: call 86 | Calls another function at address 0x86. |
| 86: lea 0x0(%rip), %rsi | Prepares %rsi for another function. |
| 8d: lea 0x0(%rip), %rdi | Prepares %rdi for the same function. |
| 94: call 99 | Calls function at 0x99. |
|  |  |
| 99: lea 0x0(%rip), %rsi | Loads argument into %rsi. |
| a0: lea 0x0(%rip), %rdi | Loads argument into %rdi. |
| a7: call ac | Calls function at 0xac. |
| ac: lea 0x0(%rip), %rsi | Loads argument into %rsi. |
| b3: lea 0x0(%rip), %rdi | |  | | --- | |  |  |  | | --- | | Loads argument into %rdi. | |
| ba: call bf | Calls function at 0xbf. |
| bf: mov %rax, %rdx | Moves return value from %rax to %rdx for next use. |
|  |  |
| c2: mov 0x0(%rip), %eax | Loads a value into %eax from memory. |
| c8: mov %eax, %esi | Moves %eax value into %esi. |
| ca: mov %rdx, %rdi | Moves %rdx value into %rdi for function call. |
| cd: call d2 | Calls function at address 0xd2. |
| d2: mov %rax, %rdx | Stores return value into %rdx. |
| d5: mov 0x0(%rip), %rax | Loads another value into %rax. |
|  |  |
| dc: mov %rax, %rsi | Moves %rax into %rsi. |
| df: mov %rdx, %rdi | Moves %rdx into %rdi. |
| e2: call e7 | Calls function at 0xe7. |
| e7: mov 0x0(%rip), %eax | Loads value from memory into %eax. |
| ed: cmp $0x1, %eax | Compares value in %eax to 1. |
| f0: jne f9 | Jumps to 0xf9 if values are not equal. |
| f2: call f7 | Calls function at 0xf7. |
| f7: jmp 109 | Unconditionally jumps to 0x109. |
| f9: mov 0x0(%rip), %eax | Loads value into %eax from memory. |
| ff: cmp $0x2, %eax | Compares that value to 2. |
| 102: jne 109 | If not equal, jump to 0x109. |
| 104: call 109 | Calls function at 0x109. |
| 109: mov 0x0(%rip), %eax | Loads another value into %eax. |
| 10f: cmp $0x3, %eax | |  | | --- | |  |  |  | | --- | | Compares value to 3. | |
|  |  |
| 112: je 119 | If equal, jump to 0x119. |
| 114: jmp 4d | Otherwise, jump back to 0x4d. |
| 119: mov $0x0, %eax | Set %eax to 0 to return 0 from main. |
| 11e: pop %rbp | Restore previous base pointer. |
|  |  |
| 11f: ret | Return from the main function. |

### ChangeCustomerChoice function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 42d: push %rbp | Saves the caller's base pointer onto the stack, preparing to create a new stack frame. |
| 42e: mov %rsp, %rbp | Sets up a new stack frame by copying the stack pointer into the base pointer register. |
| 431: lea 0x0(%rip), %rsi | Loads the address of a string (likely a menu heading) into register %rsi, the second argument for a function. |
| 438: lea 0x0(%rip), %rdi | Loads another address (possibly a format string or a label) into %rdi, the first function argument. |
| 43f: call 444 | Calls a function — likely printf() or a wrapper that displays the menu heading to the user. |
|  |  |
| 444: lea 0x0(%rip), %rsi | Loads the next prompt or string into %rsi. |
| 44b: lea 0x0(%rip), %rdi | Loads another format string or label into %rdi. |
| 452: call 457 | Displays another part of the menu or option label likely option 1. |
| 457: lea 0x0(%rip), %rsi | Loads the third string into %rsi, probably for option 2. |
| 45e: lea 0x0(%rip), %rdi | Loads the matching label or argument into %rdi. |
| 465: call 46a | Calls a function to display menu option 2. |
| 46a: lea 0x0(%rip), %rsi | Loads the fourth menu option string into %rsi. |
| 471: lea 0x0(%rip), %rdi | Loads label for the fourth option into %rdi. |
| 478: call 47d | Displays option 3 or final menu instructions. |
| 47d: mov 0x0(%rip), %eax | Loads the user’s menu selection from a memory-mapped location into %eax. |
| 483: cmp $0x1, %eax | Compares the selection to 1 — checking if user chose Option 1. |
| 486: jne 496 | If the value is not 1, jump to the next comparison. |
| 488: mov 0x0(%rip), %eax | Loads a value related to Option 1. |
|  |  |
| 48e: mov %eax, 0x0(%rip) | Stores that new name or selection into the global or static variable. |
| 494: jmp 4f8 | Skips remaining comparisons and goes to function exit. |
| 496: mov 0x0(%rip), %eax | Loads current selection again for Option 2. |
| 49c: cmp $0x2, %eax | Compares if the selection is 2. |
| 49f: jne 4af | Jumps to next block if it’s not 2. |
| 4a1: mov 0x0(%rip), %eax | Loads Option 2 input. |
| 4a7: mov %eax, 0x0(%rip) | Saves the new email address to memory. |
| 4ad: jmp 4f8 | Jumps to exit. |
| 4af: mov 0x0(%rip), %eax | Loads selection again for Option 3. |
| 4b5: cmp $0x3, %eax | Checks if the selection is 3. |
| 4b8: jne 4c8 | If not, jump to next section. |
| 4ba: mov 0x0(%rip), %eax | Loads phone number or related field. |
| 4c0: mov %eax, 0x0(%rip) | Saves the updated phone number. |
| 4c6: jmp 4f8 | Go to function exit. |
| 4c8: mov 0x0(%rip), %eax | Loads selection again for Option 4. |
| 4ce: cmp $0x4, %eax | Checks if user picked Option 4. |
| 4d1: jne 4e1 | Jumps to final option check if not 4. |
| 4d3: mov 0x0(%rip), %eax | Loads address field value. |
|  |  |
| 4d9: mov %eax, 0x0(%rip) | Updates stored address field. |
| 4df: jmp 4f8 | Exit function. |
| 4e1: mov 0x0(%rip), %eax | Last check: load value for Option 5. |
| 4e7: cmp $0x5, %eax | Checks if user selected Option 5. |
| 4ea: jne 4f8 | If not 5, skip update and jump to function end. |
| 4ec: mov 0x0(%rip), %eax | Load new password or password hash. |
| 4f2: mov %eax, 0x0(%rip) | Save password to memory. |
| 4f8: nop | No operation; often used for alignment or padding. |
| 4f9: pop %rbp | Restores the previous base pointer, tearing down the stack frame. |
| 4fa: ret | Returns control to the caller (likely main). |

### CheckUserPermissonAccess Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 120: push %rbp | Save base pointer to create a new stack frame. |
| 121: mov %rsp, %rbp | Establishes the new base pointer. |
| 124: push %rbx | Preserve the value in %rbx (used for return value logic). |
| 125: sub $0x48, %rsp | Allocate 72 bytes of space on the stack for local variables. |
|  |  |
| 129: mov %fs:0x28, %rax | Read thread-local storage to detect buffer overflows. |
| 132: mov %rax, -0x18(%rbp) | Store the stack canary on the stack. |
| 136: xor %eax, %eax | Clear the return register %eax to 0. |
| 138: lea -0x45(%rbp), %rax | Load the address of a local buffer (user input). |
| 13c: mov %rax, %rdi | Prepare the buffer as the destination for user input. |
| 13f: call 144 | Call a function to read or initialize user input. |
| 144: lea -0x45(%rbp), %rdx | Load pointer to buffer as second parameter. |
| 148: lea -0x40(%rbp), %rax | Prepare storage for result or output. |
| 14c: lea ..., %rsi | Load message or format string into %rsi. |
| 153: mov %rax, %rdi | Set the destination pointer. |
| 156: call 15b | Likely calls scanf or a similar input function. |
| 15b: lea -0x45(%rbp), %rax | Load user input buffer again. |
| 15f: mov %rax, %rdi | Pass it to a validation function. |
| 162: call 167 | Likely calls a function to sanitize or validate the input. |
| 167: movl $0x0, -0x44(%rbp) | Set a local permission flag to 0 |
| 16e: lea ..., %rsi | Load an informative string. |
| 175: lea ..., %rdi | Load label or format for display. |
| 17c: call 181 | Display permission checking info to user. |
| 181: lea ..., %rsi | Load more detail text. |
|  |  |
| 188: lea ..., %rdi | Load format or label. |
| 18f: call 194 | Continue displaying permission detail. |
| 194: lea ..., %rsi | Load more content |
| 19b: lea ..., %rdi | Load destination string or label. |
| 1a2: call 1a7 | Display another message. |
| 1a7: lea -0x40(%rbp), %rax | Load address of a result buffer or user input. |
| 1ab: mov %rax, %rsi | Move it to second argument. |
| 1ae: lea ..., %rdi | Load access control list filename or database. |
| 1b5: call 1ba | Attempt to verify user against access control list. |
| 1ba: lea -0x40(%rbp), %rax | Re-load buffer |
| 1be: lea ..., %rsi | Load comparison value |
| 1c5: mov %rax, %rdi | Prepare arguments for comparison. |
| 1c8: call 1cd | Compare credentials. |
| 1cd: mov %eax, -0x44(%rbp) | Store result of comparison (0 = no match). |
|  |  |
| 1d0: cmpl $0x0, -0x44(%rbp) | Check if result is zero (meaning access denied). |
| 1d4: jne 1dd | Jump if credentials matched (non-zero result). |
| 1d6: mov $0x1, %ebx | Set return value in %ebx to 1 (limited permission). |
| 1db: jmp 1e2 | Skip next instruction. |
| 1dd: mov $0x2, %ebx | Set return value to 2 (full permission). |
| 1e2: lea -0x40(%rbp), %rax | Reload buffer (user string or result). |
| 1e6: mov %rax, %rdi | Set up for logging or confirmation call. |
| 1e9: call 1ee | Log access level granted. |
| 1ee: mov %ebx, %eax | Copy permission level into return register. |
| 1f0: mov -0x18(%rbp), %rcx | Load original canary value for stack protection. |
| 1f4: xor %fs:0x28, %rcx | Check if canary was modified (detect stack buffer overflow). |
| 1fd: je 23a | If canary matches, jump to return block. |
| 1ff: jmp 235 | Otherwise, jump to cleanup routine. |
|  |  |
| 201–234: | These call various cleanup handlers: destroy sensitive memory, log user activity, close buffers. |
| 23a: add $0x48, %rsp | Deallocate the stack space for local variables. |
| 23e: pop %rbx | Restore preserved %rbx. |
| 23f: pop %rbp | Restore the base pointer. |
| 240: ret | Return to caller (likely main) with status in %eax. |

### DisplayInfo Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| 241: push %rbp | Save the previous base pointer to set up a new stack frame. |
| 242: mov %rsp, %rbp | Set the new base pointer for the current stack frame. |
| 4ff: sub $0x10, %rsp | Allocate 16 bytes of space on the stack for local variables. |
| 503: mov %edi, -0x4(%rbp) | Store the first function argument (\_\_initialize\_p) at -4(%rbp). |
| 506: mov %esi, -0x8(%rbp) | Store the second function argument (\_\_priority) at -8(%rbp). |
|  |  |
| 509: cmpl $0x1, -0x4(%rbp) | Check if \_\_initialize\_p == 1. This confirms that initialization is requested. |
| 50d: jne 541 | If not 1, skip initialization and jump to the end (ret). |
| 50f: cmpl $0xffff, -0x8(%rbp) | Check if the priority is 0xffff. |
| 516: jne 541 | If not 0xffff, skip initialization. |
| 518: lea 0x0(%rip), %rdi | Load the address of a global constructor function or object into %rdi. |
| 51f: call 524 | Call that constructor |
| 524: lea 0x0(%rip), %rdx | Load another global data pointer |
| 52b: lea 0x0(%rip), %rsi | Load another object reference into %rsi. |
| 532: mov 0x0(%rip), %rax | Load a pointer from memory |
| 539: mov %rax, %rdi | Move the pointer into %rdi for function call argument. |
| 53c: call 541 | Call the constructor or initializer using that pointer. |
| 541: nop | No operation — used for padding or alignment. |
| 542: leave | Restore the previous stack frame. |
| 543: ret | Return from the function. Initialization complete. |