**PSP2 Design Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Juan Alberto Gutierrez Canto | Date | 26/06/2016 |
| Program | List sort | Program # | 8A |
| Instructor | Jorge Rafael Aguilar Cisneros | Language | C++ |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective design review |
| General | * Review the entire program for each checklist category; do not attempt to review for more than one category at a time! * As you complete each review step, check off that item in the box at the right. * Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Complete | Verify that the design covers all of the applicable requirements.   * All specified outputs are produced. * All needed inputs are furnished. * All required includes are stated. | MB  X  X  X | B | R | M |
| External Limits | Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  |  |  |  |
| Logic | * Verify that program sequencing is proper.   Stacks, lists, and so on are in the proper order.  Recursion unwinds properly.   * Verify that all loops are properly initiated, incremented, and terminated. * Examine each conditional statement and verify all cases. | X  X | X |  |  |
| Internal Limits | Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  |  |  |  |
| Special Cases | * Check all special cases. * Ensure proper operation with empty, full, minimum, maximum, negative, and ero values for all variables. * Protect against out-of-limits, overflow, and underflow conditions. * Ensure “impossible” conditions are absolutely impossible. * Handle all possible incorrect or error conditions. | X  X  X  X | X |  |  |
| Functional Use | * Verify that all functions, procedures, or methods are fully understood and properly used. * Verify that all externally referenced abstractions are precisely defined. | X  X |  |  |  |
| System Considerations | * Verify that the program does not cause system limits to be exceeded. * Verify that all security-sensitive data are from trusted sources. * Verify that all safety conditions conform to the safety specifications. | X  X  X |  |  |  |
| Names | Verify that   * all special names are clear, defined, and authenticated * the scopes of all variables and parameters are self-evident or defined * all named items are used within their declared scopes |  |  | X  X  X |  |
| Standards | Ensure that the design conforms to all applicable design standards. | X |  |  |  |