Matrix Functions NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Understand binary, hexadecimal & decimal bases
* Input (numeric & string)
* Looping
* Branching
* Print formatting

|  |  |  |  |
| --- | --- | --- | --- |
| **OBJECTIVE** | **DESCRIPTION** | **RESULTS EXPECTED** | **POINTS\*** |
| **0. Pseudocode** | Pseudocode provided is clear, easy to follow and allows the program to follow program logic |  | **/20** |
| **1. Accuracy** | Output is printed clearly & neatly |  | **/10** |
|  | Matrix 1 (from data.txt) | Not reflexive  Not Symmetric  Not transitive  Transitive Closure: | **/5**  **/5**  **/5**  **/5** |
|  | Matrix 2 (from data.txt) | Reflexive  Not symmetric  Not transitive  Transitive Closure: | **/5**  **/5**  **/5**  **/5** |
|  | Matrix 3 (from data.txt) | Reflexive  Symmetric  Transitive  Transitive Closure: | **/5**  **/5**  **/5**  **/5** |
|  | Screenshots of working code is included as a link in the form provided |  | **/10** |
| **2. Style** | Code is well organized  & easy to read | * Related code is put together * Indentions are properly used to improve readability * Variable names are meaningful | **/10** |
|  | Proper functions and function files are used | * .h and .cpp files are appropriately setup * pre & post comments for functions are included | **/10** |
|  | Code is well commented | * Heading information includes Name, Date & Program name * Brief explanations before blocks of related code | **/10** |
| **Bonus Feature(s)** | Code is setup to read and process an input file |  | **+20** |
|  | Prompt user to verify path | Allows user to enter the starting and ending point of a path (I,j)  Prints an appropriate message for the user | **+10** |
| **TOTAL** | ADDITIONAL COMMENTS: |  | **/130** |

**POINTS\***

Points assigned are partially subjective, but you can expect points assigned as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 - 3 | 4 | 5 (5 point scale) |
| 0 | 1-5 | 6 – 7 | 8-9 | 10 (10 point scale) |
| 0 | 1-10 | 12-14 | 15-19 | 20 (20 point scale) |
| No evidence of meeting this requirement | Evidence of requirement, but working at a substandard level  (code is present, but does not compile, for example) | Evidence of requirement, partially working  (code is present, some values work, but other values cause an abend for example) | Requirement is predominantly met, but has some small issues  (code is present and runs, but output is not completely correct, for example) | Requirement is fully met and functioning |