**QUESTION 1**

1. How much of the total disc space is available for user data on a RAID 1 filesystem?

|  |  |  |
| --- | --- | --- |
|  | a. | 25% |
|  | b. | 50% |
|  | c. | 75% |
|  | d. | 100% |

**1 points**

**QUESTION 2**

1. How much of the total disc space is available for user data on a RAID 5 filesystem that is configured with 4 disc drives?

|  |  |  |
| --- | --- | --- |
|  | a. | 25% |
|  | b. | 50% |
|  | c. | 75% |
|  | d. | 100% |

**1 points**

**QUESTION 3**

1. How much of the total disc space is available for user data on a RAID 0 filesystem that has 3 discs that each contain 1TB?

|  |  |  |
| --- | --- | --- |
|  | a. | 1 TB |
|  | b. | 2 TB |
|  | c. | 3 TB |
|  | d. | 4 TB |
|  | e. | 5.7762996 TB |

**1 points**

**QUESTION 4**

1. When one parity bit is added to each byte of memory in a computer, what can it be used for?

|  |  |  |
| --- | --- | --- |
|  | a. | To achieve a type of fairness and equality for all. |
|  | b. | To detect and correct single bit errors. |
|  | c. | To detect any even number of bit-errors. |
|  | d. | To detect any odd number of bit-errors |

**1 points**

**QUESTION 5**

1. What can the Hamming code discussed in lecture be used for?

|  |  |  |
| --- | --- | --- |
|  | a. | To detect and correct any even number of bit-errors. |
|  | b. | To detect and correct single bit errors. |
|  | c. | To detect and correct any odd number of bit-errors |
|  | d. | To detect and correct up to two bit errors. |

**1 points**

**QUESTION 6**

1. How many Hamming Code parity bits are needed for 120 data bits?

|  |  |  |
| --- | --- | --- |
|  | a. | 3 |
|  | b. | 5 |
|  | c. | 7 |
|  | d. | 8 |

**1 points**

**QUESTION 7**

1. Given the following memory dump:

 00002600: 93 05 00 00 13 06 00 00  93 06 00 00 13 07 00 00 \*................\*  
 00002610: 93 07 00 00 93 08 d0 05  73 00 00 00 63 54 05 02 \*........s...cT..\*  
 00002620: 13 01 01 ff 23 24 81 00  13 04 05 00 23 26 11 00 \*....#$......#&..\*  
 00002630: 33 04 80 40 97 00 00 00  e7 80 40 01 23 20 85 00 \*3..@......@.# ..\*  
 00002640: 6f 00 00 00 6f 00 00 00  b7 87 00 00 03 a5 07 43 \*o...o..........C\*  
 00002650: 67 80 00 00 00 00 00 00  76 61 6c 3d 00 00 00 00 \*g.......val=....\*  
 00002660: 00 00 00 00 80 84 2e 41  1f 85 45 41 80 40 9a 44 \*.......A..EA.@.D\*  
 00002670: 4f 11 f3 c3 6e 8a 67 41  20 1b 00 00 20 1b 00 00 \*O...n.gA ... ...\*  
 00002680: 44 1b 00 00 14 1b 00 00  14 1b 00 00 04 1c 00 00 \*D...............\*  
 00002690: 44 1b 00 00 14 1b 00 00  04 1c 00 00 14 1b 00 00 \*D...............\*  
 000026a0: 44 1b 00 00 10 1b 00 00  10 1b 00 00 10 1b 00 00 \*D...............\*  
 000026b0: 04 1c 00 00 54 1f 00 00  54 1f 00 00 d4 1f 00 00 \*....T...T.......\*  
 000026c0: 4c 1f 00 00 4c 1f 00 00  34 20 00 00 d4 1f 00 00 \*L...L...4 ......\*  
 000026d0: 4c 1f 00 00 34 20 00 00  4c 1f 00 00 d4 1f 00 00 \*L...4 ..L.......\*  
 000026e0: 48 1f 00 00 48 1f 00 00  48 1f 00 00 34 20 00 00 \*H...H...H...4 ..\*  
 000026f0: 00 01 02 02 03 03 03 03  04 04 04 04 04 04 04 04 \*................\*

What is the 32-bit BIG-Endian value stored at address 0x00002658 ?

|  |  |  |
| --- | --- | --- |
|  | a. | 0x76616c3d |
|  | b. | 42 |
|  | c. | 0x3d6c6176 |
|  | d. | 0x7661 |

**1 points**

**QUESTION 8**

1. Given the following memory dump:  
     
   00002600: 93 05 00 00 13 06 00 00  93 06 00 00 13 07 00 00 \*................\*  
   00002610: 93 07 00 00 93 08 d0 05  73 00 00 00 63 54 05 02 \*........s...cT..\*  
   00002620: 13 01 01 ff 23 24 81 00  13 04 05 00 23 26 11 00 \*....#$......#&..\*  
   00002630: 33 04 80 40 97 00 00 00  e7 80 40 01 23 20 85 00 \*3..@......@.# ..\*  
   00002640: 6f 00 00 00 6f 00 00 00  b7 87 00 00 03 a5 07 43 \*o...o..........C\*  
   00002650: 67 80 00 00 00 00 00 00  76 61 6c 3d 00 00 00 00 \*g.......val=....\*  
   00002660: 00 00 00 00 80 84 2e 41  1f 85 45 41 80 40 9a 44 \*.......A..EA.@.D\*  
   00002670: 4f 11 f3 c3 6e 8a 67 41  20 1b 00 00 20 1b 00 00 \*O...n.gA ... ...\*  
   00002680: 44 1b 00 00 14 1b 00 00  14 1b 00 00 04 1c 00 00 \*D...............\*  
   00002690: 44 1b 00 00 14 1b 00 00  04 1c 00 00 14 1b 00 00 \*D...............\*  
   000026a0: 44 1b 00 00 10 1b 00 00  10 1b 00 00 10 1b 00 00 \*D...............\*  
   000026b0: 04 1c 00 00 54 1f 00 00  54 1f 00 00 d4 1f 00 00 \*....T...T.......\*  
   000026c0: 4c 1f 00 00 4c 1f 00 00  34 20 00 00 d4 1f 00 00 \*L...L...4 ......\*  
   000026d0: 4c 1f 00 00 34 20 00 00  4c 1f 00 00 d4 1f 00 00 \*L...4 ..L.......\*  
   000026e0: 48 1f 00 00 48 1f 00 00  48 1f 00 00 34 20 00 00 \*H...H...H...4 ..\*  
   000026f0: 00 01 02 02 03 03 03 03  04 04 04 04 04 04 04 04 \*................\*  
     
   What is the 32-bit LITTLE-Endian value stored at address 0x00002658 ?

|  |  |  |
| --- | --- | --- |
|  | a. | 0x76616c3d |
|  | b. | 42 |
|  | c. | 0x3d6c6176 |
|  | d. | 0x7661 |

**1 points**

**QUESTION 9**

1. Read Appendix D of the RVALP book draft (on blackboard under the content tab).  Which one of the following decimal numbers can NOT be represented accurately using the 32-bit IEEE-764 floating point format?

|  |  |  |
| --- | --- | --- |
|  | a. | 0.0019531250 |
|  | b. | 0.25 |
|  | c. | 0.1 |
|  | d. | −2.625 |

**1 points**

**QUESTION 10**

1. Read this we page: https://en.wikipedia.org/wiki/Luhn\_algorithm

What is the check-digit after applying the Luhn algorithm to the number 7992739874?

|  |  |  |
| --- | --- | --- |
|  | a. | 3 |
|  | b. | 5 |
|  | c. | 7 |
|  | d. | 9 |

For question 6. How to solve for r

[](https://cdn.discordapp.com/attachments/624280632587190296/651501705757720576/unknown.png)