You are expected to solve homework problems individually. If needed, you may seek help from your friends. However, do not copy. Show all steps with your solutions for full credit.

**Name: / 50**

1. (10 points) Complete the following table

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | Base10 |  |  | Base2 |  |  | Base16 |  |  |
|  |  |  |  |  | 11110011 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | 172 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | F1 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | 145 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | 11100010 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | 36 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | C3 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | 49 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | 01001111 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | 123 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

1. ( 5 points)The following decimal numbers are to be stored in a 6-bit 2’s complement format. Show how they are stored
   1. +14
   2. -20
   3. +37
   4. +25
   5. -45
2. ( 5 points) What is this number?

10110012 =?10

1. (5 points) What are the overflow detection rules?
2. ( 5 points) The following 6-bit 2’s complement numbers were found in a computer. What decimal number do they represent?
   1. 010000

* 1. 011001
  2. 101011
  3. 100100
  4. 100000

1. ( 10 points) Each of the following pairs of signed (2’s complement) numbers are stored in computer (6 bits).Compute the sum as it is stored in a 6-bit computer word. Show the decimal equivalents of each operand and the sum. Indicate if there is overflow.
   1. 001010

010100

* 1. 111110

001100

* 1. 001010

011000

* 1. 101001

110001

* 1. 101010

100110

1. (10 points) For each of the following pairs of signed integers (2’s complement), subtract the second from the first. Indicate overflow where appropriate
2. 111010

000111

1. 100100

011000

1. 010001

011000

1. 010000

100100