1. **(4 points)** Consider the following 16 bits: **0011 1010 0010 1001.** Give the hexadecimal representation of these bits and interpret them as a string of 8-bit ASCII characters. Use the ASCII table on the last page of the exam.

Hexadecimal: _____ ASCII characters: _____

B this page or on the scratch pages if needed). Answer the questions on the next page.

```
#include <stdio.h>

int main()
{
      int s = 100, num, i, k, val;

      scanf("%d", &num);


      for(i=1; i<num; i=i+1)
      {
            scanf("%d",&k);

            s = s - k;

            /* CHECKPOINT A */
      }
      val = (num*(num+1))/2;

      /* CHECKPOINT B */

      printf("The number is %d!!", val-s);

      return 0;
}
```

2.

1. **(4 points)** Perform the following bitwise logical operations. Express each answer as a single hexadecimal digit.

   a) xE XOR x4 = _____ (answer in hexadecimal representation)

   b) NOT(xC OR (NOT(x5))) = _____ (answer in hexadecimal representation)

Please compute the following arithmetic operations in 8-bit 2's complement. Express your answer as an 8-bit 2's complement number. Indicate if it has an overflow by circling the corresponding YES or NO.

Part A (3 points): $00110110 + 00000100 =$ _____ Overflow? YES NO

Part B (3 points): $01101001 + 10111010 =$ _____ Overflow? YES NO

Part C (3 points): $10101101 + 10110110 =$ _____ Overflow? YES NO

Part D (3 points): $10011011 - 11001100 =$ _____ Overflow? YES NO