

ISU Programming Assessment, April 06, 2018

Name: _____ CS class _____

Put all answers in boxes. Nothing you write outside the boxes will be counted. Did you bring an eraser?

1. Write a program that gets an integer, **n**, from the user and then prints **n** repetitions of the pattern: some A's, some B's, then a C. Each new repetition has one more A and one less B than the pattern before it. The first pattern has 1 A and **n** B's. **Example:** if the number from the user is **n=4**, then the program prints:
ABBBBCAABBBCAAABBCAAAABC

```
int main(int argc, char *argv[]) {
```

```
    return 0;  
}
```

2. **Beginning, End.** Write a C program that reads from stdin one 8-bit character at a time. The program should count the letters A-F, T-Z. It should print just one number, the final count.

```
int main(int argc, char *argv[]) {
```

```
    return 0;  
}
```

3. Write the function `smallest` that is passed the address of the first node of the list. This function considers each number in the list. It finds and returns the smallest value in the list.

```
typedef struct NODE {
    int data;
    struct NODE *next;
} node_t;
node_t head;
int smallest(node_t *curr) {
```

```
}
```

4. A BST is constructed in the usual way using the node definition below. Write a function

`int greater(int val, bst_node_t *curr)` that is given a value and the head of the tree. It returns the number of nodes containing numbers larger than `val`.

```
typedef struct BST_NODE_T {
    int data;
    struct BST_NODE_T *left, *right;
} bst_node_t;
```

5. Write the function

```
int count101(int n)
```

that counts the number of occurrences of the bit pattern 101 in **n**. It should return the final count. Variable **n** is 32 bits. **Example:** Suppose **n** is just 8 bits, say **n=11010101**, then the function would return 3.