

# ISU Programming Assessment, April 27, 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 **X**'s, some **Y**'s. Each new repetition has three more **X**'s and one less **Y** than the pattern before it. The first pattern has 1 **X** and **n** **Y**'s. **Example:** if the number from the user is **n=3**, then the program prints:  
XYYYYXXXXYXXXXXXXXY

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

```
    return 0;  
}
```

2. **Getting input a character at a time.** Write a program that counts each 'A' that is preceded by a space. It should print the final total. **Example the input:** "An Ant, A dog." has 2 A's preceded by a space.

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

```
    return 0;  
}
```

3. Write the function `addOdd` that is passed the address of the first node of the list. This function considers each number in the list. It returns the sum of all the odd numbers in the list.

```
typedef struct NODE {
    int data;
    struct NODE *next;
} node_t;

int addOdd(node_t *curr) {
```

```
}
```

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

```
int sum( bst_node_t *curr)
```

that returns the sum of all the nodes in the tree.

```
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 101's in `n`. It returns the final count. **Example (8 bits):** 10101000 has 2 101 patterns.