

ISU Programming Assessment, April 13, 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 0's. Each new repetition has two more X's and one less 0 than the pattern before it. The first pattern has 1 X and n 0's. **Example:** if the number from the user is $n=4$, then the program prints:
X0000XXX000XXXXXX00XXXXXX0

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

```
    return 0;  
}
```

2. **Non-Letter.** Write a C program that reads from stdin one 8-bit character at a time. The program should find the number of non-letter characters in its input. It should print this number.

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

```
    return 0;  
}
```

3. Write the function `square` that is passed the address of the first node of the list. This function considers each number in the list. It counts each time that number is the **square** of the next number. It returns the final count.

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

```
}
```

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

```
int longest( bst_node_t *curr)
```

that returns the number of nodes in the longest path from `curr` to a leaf. Remember to count both `*curr` and the leaf.

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

5. Write the function

```
int high(int n)
```

that finds the number of 1's in the sixteen high bits and the number of 1's in the 16 low bits. It returns 1 if the high bits have more 1's than the low bits and returns 0 otherwise.