

ISU Programming Assessment, Nov 09, 2018

Name: _____ CS class _____

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

1. **Odds.** The odd numbers are 1, 3, 5, and so on. **Write** a program that gets an integer, **n**, from the user and then prints **n** repetitions of the pattern: some **X**'s then one **0**. The first pattern, (**pattern 1**), has one **X** then one **0**. Let **p** stand for 2, 3, or higher. The **p**-th pattern (**pattern p**) has **p**-th odd number *more X*'s than the pattern before it.

Example: if **n=3**, then the program will print **XOXXXXOXXXXXXXXX0**

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

```
    return 0;  
}
```

2. **Get input a character at a time.** Write a program that counts the total number of **NON** digits in the input. It should print out the only the final count.

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

```
    return 0;  
}
```

3. Write the function `replace` that is passed the address of the first node of the list. If the list is empty, then the function does nothing. The function does not change the first node. After the first node, it replaces each data with data minus the previous data.

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

```
}
```

4. A BST is constructed in the usual way using the node definition below. A "leaf" is a node that has no children. Write the function

```
int numLeaf( bst_node_t *curr)
```

that is passed a pointer to a node. It returns the number of leaf nodes in the tree determined by `curr`.

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

5. Write the function

```
int Ominus1(int n)
```

where `n` is a 32-bit int. The function returns the number of 0-bits minus the number of 1-bits in `n`.