ISU Programming Assessment, Oct 26, 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 `XO`'s then a `B`. Each new repetition has one more `XO`. The first pattern has one `XO` then a `B`. **Example:** if `n=3,` then the program will print `XOBXOXOBXOXOXOB`

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

```
      return 0;
    }
```

2. **Getting input a character at a time.** Write a program that counts any letter (lower case) in the first half of the alphabet (`'a'-'m'`). It should print out the only the final count.

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

```
      return 0;
    }
```

3. Write the function `equal` that is passed the address of the first node of the list. The function counts the number of times a number in the list is EQUAL to the next number in the list. It returns the final count.

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

```
int  equal(node_t *curr) {
```

```
    }
```

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

```
        int child2( bst_node_t *curr)
```

that returns the number of nodes that have 2 children.

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

5. Write the function

```
        int opposites(int n)
```

where `n` is a 32–bit int. The function compares each bit in the high half, $b_{31}b_{30}...b_{17}b_{16}$, with the corresponding bit in the low half, $b_{15}b_{14}...b_1b_0$.

That is, b_{16} is compared with b_0, and b_{17} is compared with b_1 and so on.

If all pairs consist of opposites (1 and 0 **or** 0 and 1) the function returns 1. Otherwise it returns 0.