

# ISU Programming Assessment, May 03, 2019

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** patterns. Each pattern consists of some **A**'s then some **B**'s. The first pattern is one **A** followed by one **B**. Each new pattern uses the next two fibonacci numbers for the number of **A**'s and the number of **B**'s. Fibonacci: 1, 1, 2, 3, 5, 8, 13, ... **Example:** if **n=3**, then the program will print **ABAABBBAAAAABBBBBBBB**.

```
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 two character sequences consisting of a lower case letter followed by second lower case letter that is larger than the first. It should print out the only the final count. Example: if the input is "bet dog" the count is 3.

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

```
    return 0;  
}
```

3. Write the function `sumEveryOther` that is passed address 0 or the address of the head node of the list. It finds the sum of the data in every other node of the list. It returns this value.

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

int sumEveryOther(node_t *curr) {
```

```
}
```

4. A BST is constructed in the usual way using the node definition below. **Write** the function `int countNodeWLC( bst_node_t *curr)` that is passed address 0 or the address of the root node of the BST. It returns the number of nodes that have a left child.

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

5. Write the function

```
int rotate(unsigned int n)
```

where in bits,  $n = b_{31}b_{30}...b_0$ . The function changes  $n$  so that  $n = b_0b_{31}b_{30}...b_1$ . This new value of  $n$  is returned.

