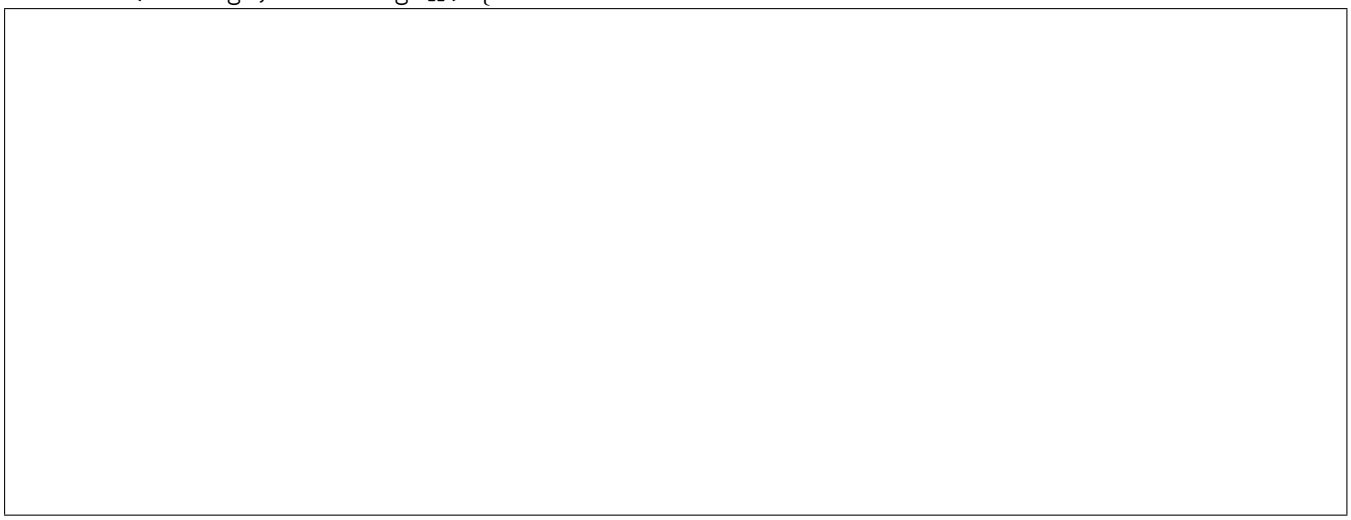
ISU Programming Assessment, April 05, 2019

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. The first pattern is `n` `A`'s. The next pattern is `n-1` `B`'s. Each new pattern has one fewer letter than the previous. The `A`–patterns and `B`–patterns alternate. **Example:** if `n=4`, then the program will print `AAAABBBAAB`.

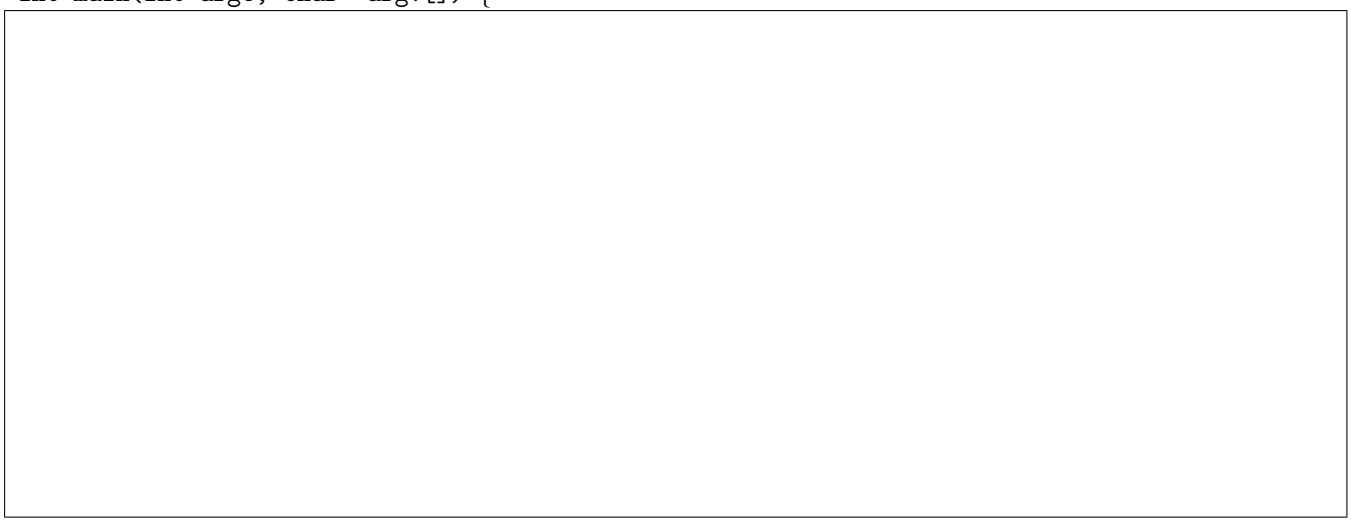
```
int main(int argc, char *argv[]) {
```



```
  return 0;
}
```

2. **Get input a character at a time.** Write a program that counts the total number of sequences that consist of a lower case letter followed by a lower case `n`. That is, count the total of all occurrences of all sequences: `an`, `bn`, `cn`, ... `zn`. It should print out the only the final count.

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



```
  return 0;
}
```

3. Write the function `addFive` that is passed the address of the first node of the list. The function finds the sum of the first five numbers in the list. If there are fewer than five numbers, it finds the sum of the numbers that are there. It returns the sum.

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

int  addFive(node_t *curr) {
```

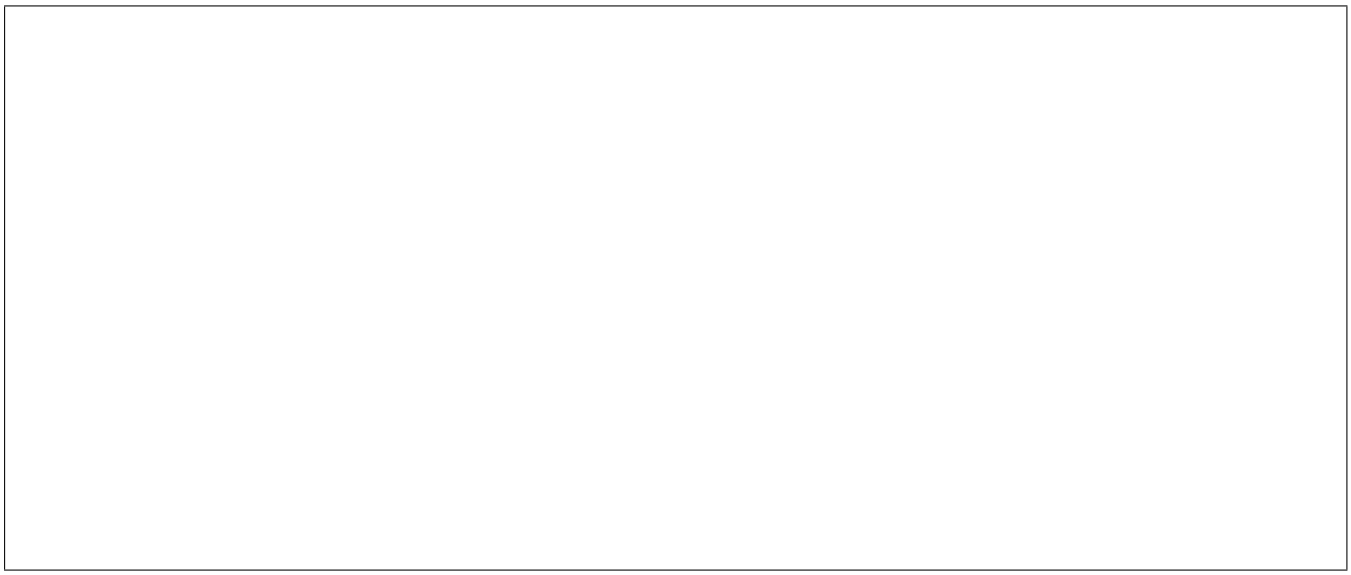
```
}
```

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

```
int NwithRC( bst_node_t *curr)
```

that is passed the address of the head node of the BST. It counts and returns the number of nodes in the BST that have a right child.

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



5. Write the function

```
int highCount00(int n)
```

where `n` is a 32–bit int. The function counts and returns the number of 00 bit sequences in the upper half of `n`, that is, $b_{31}b_{30}...b_{16}$. **Example:** the upper half of the 8 bit sequence `10001000` contains two 00 sequences.