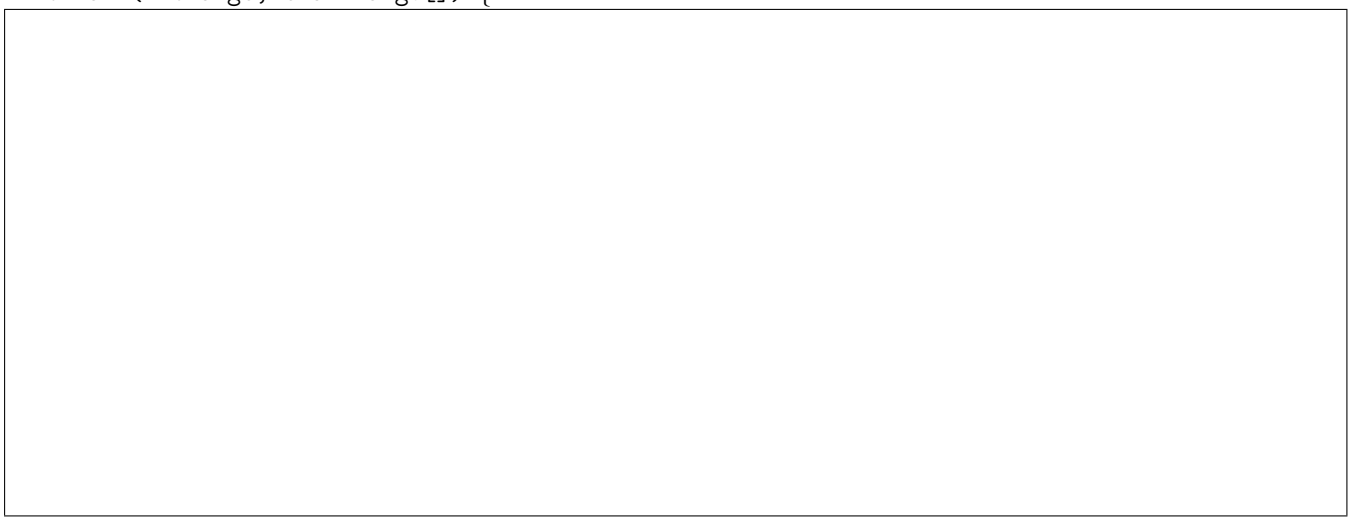
ISU Programming Assessment, March 9, 2018

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 C program that gets a number, `n`, from the user. The program will print **ONE** line that will consist of `n` patterns. Each pattern has one `A` that starts the pattern. The rest of the pattern is `B`'s and `C`'s. Each group of `C`'s follows a `B`. The first pattern is: `ABC`. The next pattern is: `ABCCBCC`. Each new pattern has one more `B` than the last and each group of `C`'s after a `B` has one more `C` than in the previous pattern. Example: n=3: `ABCABCCBCCABCCCBCCCBCCC`

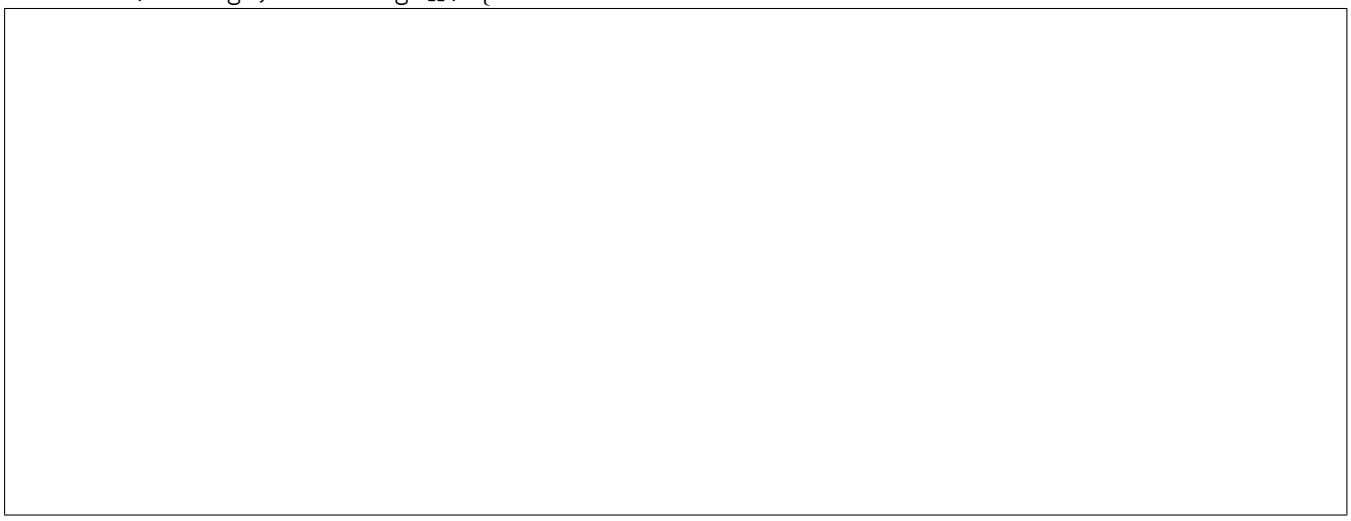
```
int main(int argc, char *argv[]) {
```



```
    return 0;
}
```

2. **Average Character. Write** a C program that reads from stdin one 8-bit character at a time. The program should find the sum of all the character codes and the number of characters. It should print just one value: the average character code of its input.

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

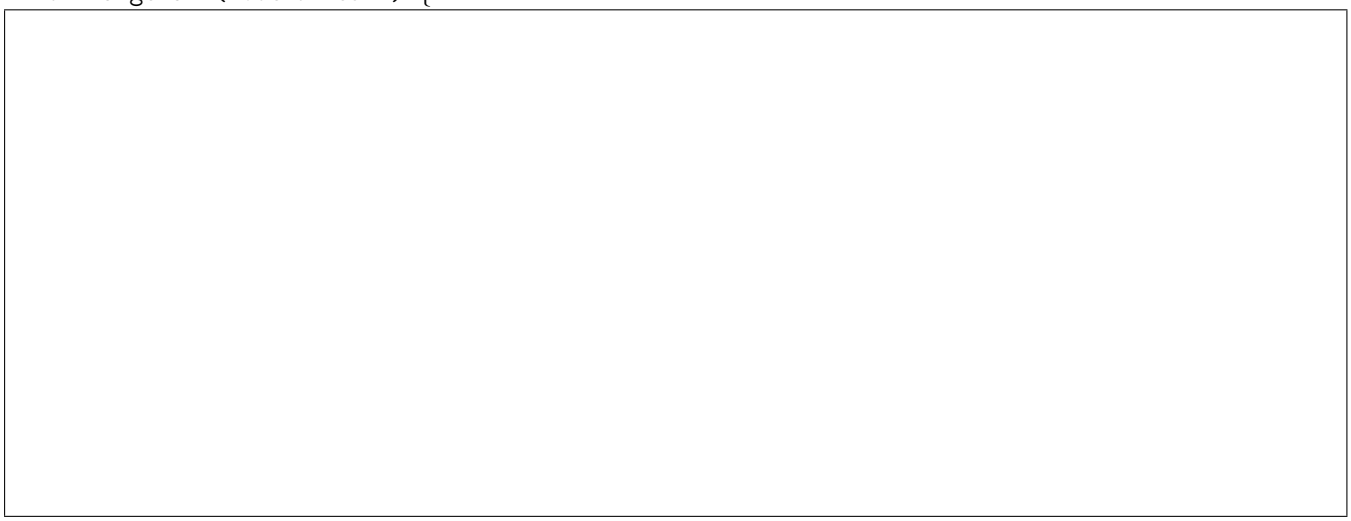


```
    return 0;
}
```

3. Write the function `largePair` 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 **LARGER** than the next number. It returns the final count.

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

int  largePair(node_t *curr) {
```



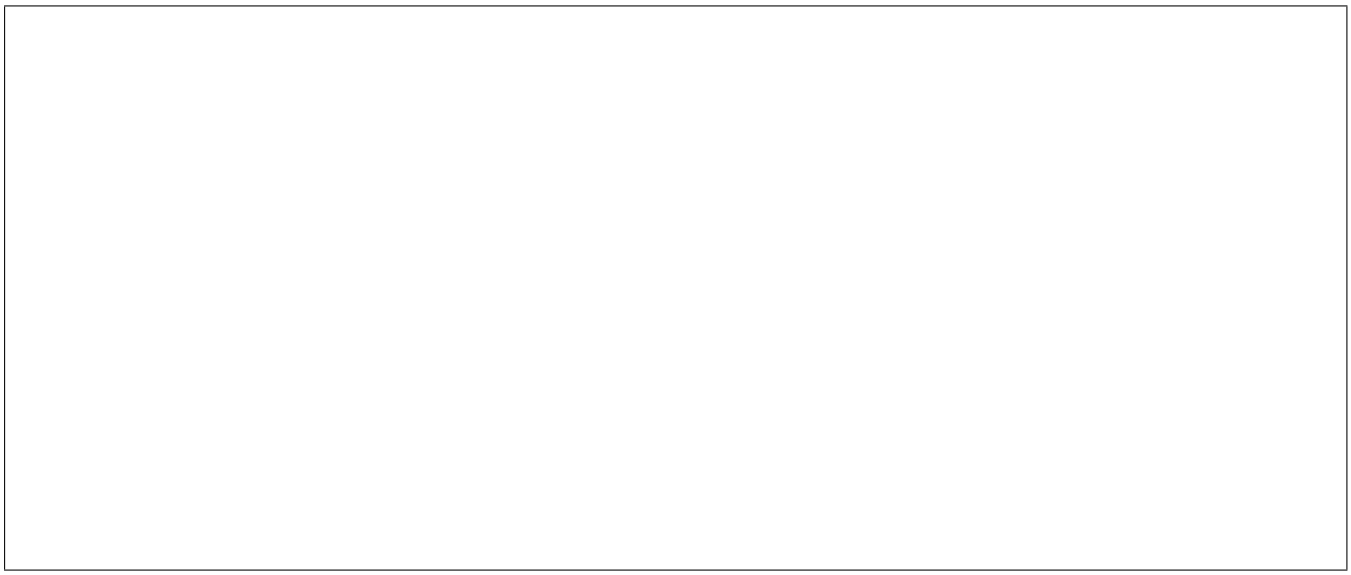
```
}
```

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

`bst_node_t *search(int sv, bst_node_t *curr)`

that returns the address of the node containing the search value, `sv`, or returns 0 if `sv` is not in the tree. **REQUIREMENT:** Do NOT solve this with a traversal.

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



5. Write the function

```
int sum(int n)
```

that breaks `n` into 8 pieces of 4 bits each. It returns the sum of the 8 pieces. **Example (for two pieces):** `10100111`. The pieces are 10 and 7. The sum, 17, should be returned by the function.