Programming Design Worksheet - Redfield
for CS1310 (programs 2-7) and CS1311 (programs 1-6)

Copy this file. Type and past images to create new documents for designs.
Print it for class (if you must miss, submit one file to Designs).

First name **Davide** Last name **Russillo**

Design for program name **Guessing Tree**

DATA

Variables needed in **WORDS for main** and **globally**
structure definition for a node, containing value, left, and right variables
pointer to root node of tree

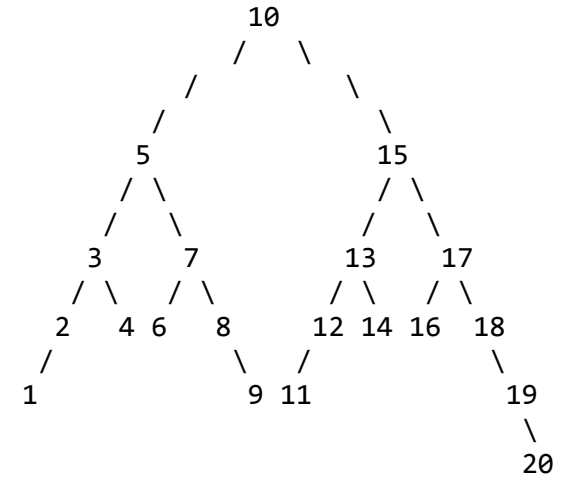
C ***DECLARATIONS*** for main & global
global:

```
struct Node {
      int value;
      struct Node *left;
      struct Node *right;
};
```

main:

```
struct Node *root = malloc(sizeof(struct Node));
```

(STARTING TicTacToe:put image; or draw: Insert, Drawing; or put at end of the file)
draw in RAM with possible values



Algorithm to PSEUDOCODE level for each function
(*remember to indent under* if, switch, while, do-while, for)

main:
```
call init_tree on root
call populate_tree on root

print think of a number between 1 and 20 and I will try to guess it.
      Press enter to start …
while get char input is not newline, do nothing

call take_user_input on root
```

other functions (bold the names): (put them before main in the program!)

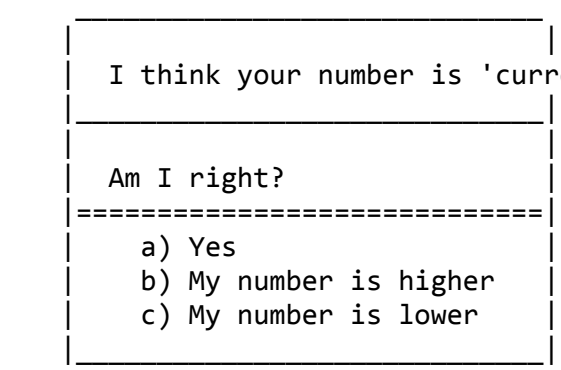
void init_tree(struct Node *root)
```
      init_tree_helper(struct Node *current, int current_depth)
            allocate memory for left node of current
            allocate memory for right node of current
            if the current depth is less than 2
                  call init_tree_helper on left with current_depth + 1
                  call init_tree_helper on right with current_depth + 1
      call init_tree_helper on root
      allocate memory for left of left of left of left of current
      allocate memory for left of right of right of right of current
      allocate memory for right of left of left of left of current
      allocate memory for right of right of right of right of current
      allocate memory for right of right of right of right of right of
                                                                  current
```

void populate_tree(struct Node *root)
```
      populate_tree_helper(struct Node *root, int difference, int set_value)
            set current value to set_value
            set difference to difference / 2
            if difference is 0
                  set difference to 1
            if current left is not uninitialized
                  call populate_tree_helper with current left, difference,
                                                and set_value - difference

            if current right is not uninitialized
                  call populate_tree_helper with current left, difference,
                                                and set_value + difference
      call populate_tree_helper on root
```

void take_user_input(struct Node *current)
```
      set choice character to ' '
      set valid choice boolean to false
```

print




```
while not valid choice
      set choice to character input
      set valid choice to character input is equal to a, b, or c
      if not valid_choice
            print invalid answer, try again!
if choice is c
      if current left is not uninitialized
            call take_user_input on current left
      else
            print Invalid number!
else if choice is b
      if current right is not unitialized
            call take_user_input on current right
      else
            print Invalid number!
else
      print I guessed right! Your number is 'current value'
```