• Task 1—Cube setup, board drawing, cube shaking. Design your data structure for the cubes and board. It will help to group related data into sensible structures rather than pass a dozen parameters around. As usual, no global variables. Set up and shuffle the cubes. You need to use a 5x5 board configuration.

• Task 2—Human’s turn (except for finding words on the board). Write the loop that allows the user to enter words. Reject words that have already been entered or that don’t meet the minimum word length or that aren’t in the lexicon. Add words to the graphical display and keep score.

• Task 3—Find a given word on the board. Now put your recursive talents to work in verifying that a word can be formed on the board, subject to the adjacency and nonduplication rules. You will employ recursive backtracking that “fails fast”: as soon as you realize you can’t form the word starting at a position, you move on to the next position. If a path is found, highlight the cubes that form the word.

• Task 4—Find all the words on the board (computer’s turn). Now it’s time to implement the killer computer player. Employing the power of recursion, your computer player traverses the board using an exhaustive search to find all remaining words. Be sure to use the lexicon prefix search to abandon searches down dead-end paths.

• Task 5—Loop to play many games, add polish. Once you can successfully play one game, it’s a snap to play many. With everything now working, it’s time to finish off the details. Be sure to gracefully handle all user input. Make sure your comments are thoughtful and complete. Pat yourself on the back and go eat some ice cream!

• Task 6—Make the Q a useful letter. Because the Q is largely useless unless it is adjacent to a U, the commercial version of Boggle prints Qu together on a single face of the cube. You use both letters together—a strategy that not only makes the Q more playable but also allows you to increase your score because the combination counts as two letters.

• Task 7—Play by Mouse. Since we have prepared a graphical user interface, the player could explore words by clicking on the board. Your program should track and highlight which cubes have been clicked (and thus, selected) and their order. Once the selected cubes form a valid word that hasn’t been explored before, your program should award the player and append the word to the player’s list. The selection and highlight should be cleared if 1) a word has been added to the player’s list; or 2) the current selected cubes cannot form any word

• 任务1-立方体设置，板绘制，立方体晃动。为多维数据集和电路板设计数据结构。它将有助于将相关数据分组到合理的结构中，而不是传递十几个参数。像往常一样，没有全局变量。设置并移动立方体。您需要使用5x5板配置。

• 任务2-轮到人（除了在黑板上找到单词）。编写允许用户输入单词的循环。拒绝已输入或不符合最小字长或不在词典中的单词。将单词添加到图形显示并保留分数。

• 任务3-在黑板上找到给定的单词。现在让你的递归天赋来验证一个单词是否可以在黑板上形成，并遵循邻接和非重叠规则。你将采用递归回溯法，即“快速失败”：一旦你意识到你不能从一个位置开始形成这个词，你就进入下一个位置。如果找到路径，请突出显示构成单词的立方体。

• 任务4-找到黑板上的所有单词（轮到计算机）。现在是时候实施杀手电脑播放器了。利用递归的能力，你的电脑播放器使用一个详尽的搜索来找到所有剩余的单词。请确保使用词典前缀搜索放弃在死胡同中的搜索。

• 任务5-循环玩许多游戏，添加波兰语。一旦你能成功地玩一个游戏，玩很多游戏就很容易了。现在一切都好了，是时候结束细节了。请务必优雅地处理所有用户输入。确保你的评论是周到和完整的。拍拍自己的背，去吃点冰淇淋吧！

• 任务6——把问题变成一封有用的信。因为q基本上没有用处，除非它与u相邻，所以商业版的boggle将qu一起印在立方体的一个面上。你同时使用这两个字母-这一策略不仅使q更容易玩，而且还允许你增加你的分数，因为组合算作两个字母。

• 任务7-鼠标播放。由于我们已经准备了一个图形用户界面，玩家可以通过点击黑板来浏览单词。您的程序应该跟踪并突出显示哪些多维数据集已被单击（从而被选中）及其顺序。一旦选定的多维数据集形成了一个以前没有被探索过的有效单词，您的程序就应该奖励玩家并将该单词附加到玩家列表中。如果1）已将一个单词添加到播放机列表中；或2）当前选定的多维数据集无法形成任何单词，则应清除选择和突出显示。