Topic – Sudoku

1) **Definition** - This will be a C++ based project. Our aim is to create a user-friendly Sudoku game which has a lot of flexibility and incorporates many new features such as the level of difficulty, record of high scores, a timer, the ability to check whether or not user input is correct (checker). In addition to this, there will be a bonus Hexadoku version included (Sudoku using hexadecimal values). The reason behind working on this is because all of us enjoyed solving these puzzles as children. We decided that doing this project would be fun working on as well as testing it.
2) **Analysis** –

   <u>Inputs</u> –

   1) Menu 1 Options (Start game, see high scores, exit game)
   2) Menu 2 Options (whether user wants to attempt the easy, medium, hard, or bonus Hexadoku)
   3) Menu 3 Options (place number on grid, use a checker/hint, exit game)
   4) Grid Positions (position on the grid the user wants to "access")
   5) Grid Value (the number the user wants to appear on the coordinate chosen)
   6) Post-game Menu Options (play again, quit game)

   <u>Outputs</u> –

   1) The 3 menus
   2) The Grid
   3) Post-game "congratulations with the score and menu"
3) **Design** – On the next page in the form of a UML model
4) **Execution Plan** –

   <u>Dividing workload:</u>

   1) Rahul & Yaswanth – Working on the Sudoku grids
   2) Bharat & Ash – Timer & checkers
   3) Preetham – Display menus and make sure there are no errors
   4) Derek & Joseph – researching, managing high scores and Hexadoku grid

   Workload was divided keeping in mind each persons' abilities and areas of interest.

   <u>Make file</u> on the next page

   <u>Team Deadlines</u>

   April 1st – Menu and grids w/o checker
   April 15th – Checker and timer as well as placing the values in the coordinates
   April 23rd – Working out input errors

Makefile

Welcome to Sudoku!

Please select any one of options from the menu below:

1. Start Game
2. Display High Scores
3. Exit Game

Are you sure you want to exit?

1. Yes 2. No

Attempt 1: " "

Attempt 2: "...

:

Attempt n: " "

Select Difficulty:

1. Easy
2. Medium
3. Hard
4. Bonus (Hexadoku)

Sample Output of Sudoku Grid:

	1	2	3	4	5	6	7	8	9
A				9		2			
B		4						5	
C			2				3		
D	2								7
E				4	5	6			
F	6								9
G			7				8		
H		3						4	
I				2		7			

Choose the position at which you wish to place your number:

" "

Choose the number you wish to keep at " ":

UML

```
SudokuGrid
----------
- int gridValues
- char gridPosition
- int difficulty
----------
+ int gridAssembler (int difficulty)
+ int gridChoice (int gridValues,
                  char gridPosition)
```

```
SudokuChecker
----------
- int timerVal
- int checker
----------
+ int sudokuVal(int timerVal)
+ int sudokuCheck(int checker)
```

```
SudokuMenu
----------
- int MenuSelect
- int diffSelect
- int endgame
----------
+ void menuDisplay()
+ void inputValid()
+ void endDisplay()
```