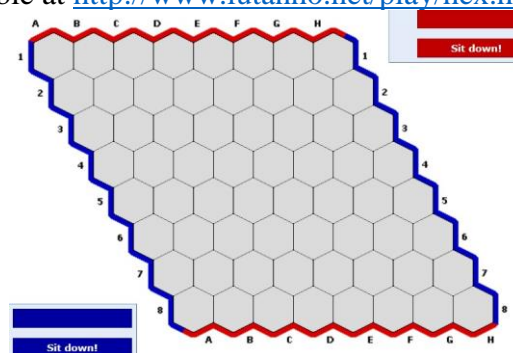


You will write a C++ program that will play the game of HEX. The game of Hex is played by two players (computer-user or user1-user2) on a two-dimensional board (2D C array) with hexagonal cells (each cell has 6 neighbors: east, west, northeast, northwest, southeast, southwest). The neighbors of F3 are G2, F2, E3, E4, F4, and G3. Each cell is either occupied by computer (user1), user (user2) or empty. The game starts with the following board for an 8x8 game:

	a	b	c	d	e	f	g	h
1	.	.	.	.	.	.	.	.
2		.	.	.	.	.	.	.
3			.	.	.	.	.	.
4				.	.	.	.	.
5					.	.	.	.
6						.	.	.
7							.	.
8								.

where '.' represents the empty cells, 'x' represents the computer(user1) cells and 'o' represents the user (user2) cells. Please see <https://www.maths.ed.ac.uk/~csangwin/hex/index.html> for the rules of the game (we will ignore the swap rule). The following is a board from <http://www.iggamecenter.com/>. There is playable game available at <http://www.lutanho.net/play/hex.html>



The players take turns to play the game. For the above board, the user can pick any cell to put the first piece. For example, if the user1 plays to cell C5 and user2 plays to letter F3 then the board becomes

	a	b	c	d	e	f	g	h
1	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.
3	.	.	.	.	.	o	.	.
4	.	.	.	.	.	.	.	.
5	.	.	x	.	.	.	.	.
6	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	.

The game continues with the objective of “The first player to form a connected path of their marks linking the opposing sides of the board marked by their color wins.” For example, in the following board, the user1 wins because he/she connects the blue (numbers) sides. The winning path is marked with capital X and O.

	a	b	c	d	e	f	g	h
1	.	.	.	.	o	.	.	.
2	.	.	.	.	o	o	.	.
3	.	.	.	.	.	o	.	.
4	.	.	o	X	X	o	X	X
5	.	.	X	o	X	X	.	.
6	.	.	X	X	o	.	.	.
7	.	.	o	.	.	.	x	.
8	.	.	o	.	.	x	.	.

Your program will do the followings:

1. When your program starts, you will ask the user for the board size. The board size can be 6x6, 7x7, 8x8, 9x9, 10x10, 11x11 or 12x12. You should check whether the input size is valid or not.
2. Ask the user if this a two player game or user versus computer game.
3. You will display the initial board. We will ignore the swap rule of the game.
4. You ask the user to make a move, get the user move from the keyboard and draw the new board. Use positions such as A 1, B 7, C 10, etc. If the move is not legal, then ask for another move.
5. The computer (or user2) makes a new move and draws the board again. You should print what move the computer chose on the screen. The move should be legal and try to make it “smart”. In other words, chose the legal move that wins the largest number of cells instead of other moves. For the smart moves, use the suggested strategies described at <https://www.maths.ed.ac.uk/~csangwin/hex/index.html>
6. The game continues until all the board is filled or one of the players wins.
7. The final result of the game should be printed on the screen showing who won and the connected cells with upper case O and X.

Notes:

- Do not use any functions from the standard C library (like `printf`)
- Your program should have only functions and no classes.
- Do not forget to indent your code and provide comments.
- Check the validity of the user input.
- **Test your programs very carefully at least with 10 different runs. For some runs, let the computer win and for others the user should win. There should be cases for illegal moves and your program should handle them appropriately.**
- You should submit your work to the Moodle page. We will post the submission rules soon which should be strictly followed.

You will continue developing your Hex program with this homework. You will add the following features to your program.

- Your program should accept commands for loading and saving the game board. For example, instead of giving the column letters, if the user enters the command  
LOAD FILE.TXT  
your program loads the game board from this file. Similarly, if the user enters the command  
SAVE FILE.TXT  
your program saves the current game status to this file. You should save and load all game related information such as user vs user or computer, board size etc.
- Your program should have at least one function that takes a parameter by reference and a constant parameter by reference.
- Your program should have at least one function that uses default arguments.
- Your program should have at least one `auto` statement and one `decltype` statement.
- Your cell states should be represented using C++11 Enum types.
- All the errors should be printed to `cerr` object and all the other text should be sent to `cout`.

The rest of the program will be the same as HW1.

Notes:

- Do not use any functions from the standard C library (like `printf`)
- Read the chapter about file input output for reading and writing text files using streams.
- Do not use anything that we did not learn in the lectures.
- Do not forget to indent your code and provide comments.
- Check the validity of the user input.
- **Test your programs very carefully at least with 10 different runs. Submit at least two saved files with toue HW.**
- You should submit your work to the moodle page.