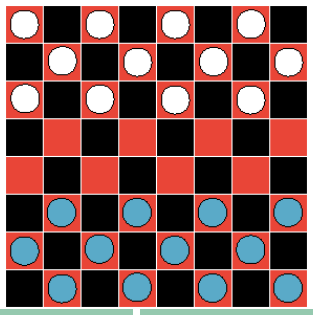
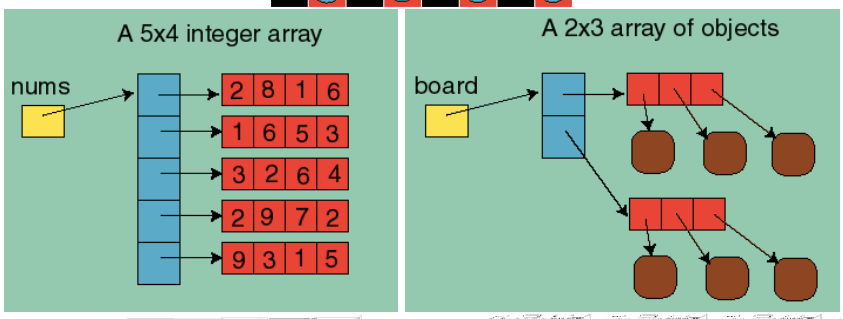
Introduction to Programming

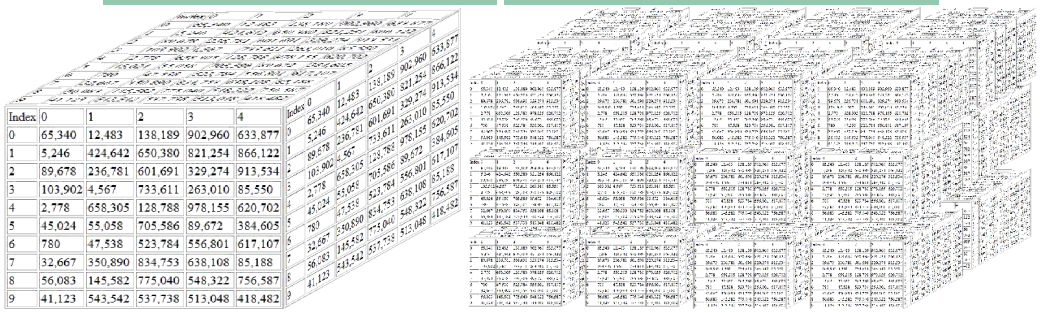
(CS200)

Shafay Shamail

Arrays, Pointer, Matrices







Index	0	1	2	3	4
0	65,340	12,483	138,189	902,960	633,877
1	5,246	424,642	650,380	821,254	866,122
2	89,678	236,781	601,691	329,274	913,534
3	103,902	4,567	733,611	263,010	85,550
4	2,778	658,305	128,788	978,155	620,702
5	45,024	55,058	705,586	89,672	384,605
6	780	47,538	523,784	556,801	617,107
7	32,667	350,890	834,753	638,108	85,188
8	56,083	145,582	775,040	548,322	756,587
9	41,123	543,542	537,738	513,048	418,482

CS 200 Lab 04 Spring 2018

Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. You put all your work onto the LMS folder designated for the lab (i.e. "Lab04") before the time of the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.
5. Any cheating case will be reported to Disciplinary Committee without any delay.
6. **NOTE: Define a class separately and its methods separately. Do not write inline code.**

Marks: Name: _____ Roll #: _____

Task 1	Q1	Q2	Q3			Total
	10	10	10			30

Task 2	Q1	Q2	Q3	Q4	Q5	Q6		Total
	5	5	5	5	5	5		30

Task 3	Q1	Q2	Q3	Q4				Total
	10	10	10	10				40

Let's Begin……

Total marks Obtained

/100

CS 200 Lab 04 Spring 2018

Task 1: (20)

You have been given a pointer to an array.

1. Write a function *reverseArray()* that takes a pointer to an array and reverses it. 10
2. Write a function *printArray()* that takes a pointer to an array and prints the array such that the number of items printed per line is not more than 10. 10
3. Write a test function that tests *reverseArray()* and *printArray()*. It should first print the original array and then the reversed array. 10

Note:

- You are not allowed to use indexing. However, you may use pointer arithmetic, involving +, -, +=, -=, ++, -- operators, to access elements of the array.
- Take the array size represented by the last two digits of your roll number. For example, if your roll number is 20101234 then the array size will be 34.
- Choose *typedef* to define data type. For example, *typedef int Data*.

STOP AND SHOW YOUR WORK TO THE TA

CS 200 Lab 04 Spring 2018

Task 2: (30)

1. Create a class called Matrix() to represent a 2D array. 5
2. Declare appropriate constructor(s) and destructor. 5
3. Write a method *transpose()* that calculates transpose of the given matrix. 5
4. Write a friend method *printMatrix()* that prints the matrix in proper format. 5
5. Overload << operator as friend so that it prints the matrix in proper format. 5
6. Test the methods *transpose()*, *printMatrix()* and *operator<<* using a *testMatrix()* function. Note that this is not the *main()* function. 5

Note:

- Transpose of a matrix is obtained by turning all the rows of a given matrix into columns and vice-versa.
- You are not allowed to use indexing. You can only use pointer arithmetic.
- You can re-use the code of Lab03 for parts 1 and 2 only.

STOP AND SHOW YOUR WORK TO THE TA

CS 200 Lab 04 Spring 2018

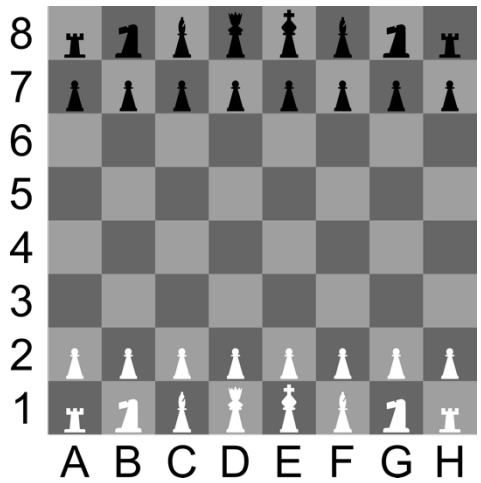
Task 3: (40)

In this task, you are required to create a class called Chess. This class will mimic some of the actions that an actual chess board performs.

There are 6 main pieces in Chess: King, Queen, Bishop, Knight, Rook and Pawn.

- The ***Rook*** moves any number of vacant squares forwards, backwards, left, or right in a straight line.
- The ***Bishop*** moves any number of vacant squares diagonally in a straight line. Consequently, a bishop stays on squares of the same color throughout a game.
- The ***Queen*** moves any number of vacant squares in any direction: forwards, backwards, left, right, or diagonally, in a straight line.
- The ***King*** moves exactly one vacant square in any direction: forwards, backwards, left, right, or diagonally.
- The ***Knight*** moves on an extended diagonal from one corner of any 2×3 rectangle of squares to the furthest opposite corner. Consequently, the knight alternates its square color each time it moves.
- The ***Pawn*** moves forward exactly one space, or optionally, two spaces when on its starting square, toward the opponent's side of the board. When there is an enemy piece one square diagonally ahead of the pawn, either left or right, then the pawn may capture that piece.

This is how a traditional chess board looks like.



CS 200 Lab 04 Spring 2018

1. You are now required to initialize the Chess class with a 2D array as your chessboard. The 6 chess pieces would be the class variables as well. Write a constructor for the class, with all the 6 pieces initialized to their starting position for both black and white pieces. Write a destructor for the class. 10
2. Write a class function that takes in the position of a pawn and outputs whether that position is a valid starting position for either a black or white pawn. (For example: B2 is valid white pawn starting position but D3 is not). You are not allowed to hard code these, use conditions and ranges to check. 10
3. Write another class function which takes in the initial position and the final position of the King and tells whether it is a valid move or not. 10
4. The Knight is the most interesting piece of chess. Write a class function, which takes in a starting position of the Knight and prints all the valid moves of that Knight. 10

STOP AND SHOW YOUR WORK TO THE TA

CS 200 Lab 04 Spring 2018

Zip your tasks into one folder with format:
YourRollNo-Lab04
example "**2001001-Lab04**" and upload on LMS before the tab is closed. You will not be given extra time.