Université d’Ottawa
Faculté de génie

École de science informatique
et génie électrique

uOttawa

L’Université canadienne
Canada’s university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

CSI 2372
Advanced Programming Concepts with C++

Final Examination

Length of Examination: 2 hour and 50 minutes **December 14th, 2022, 9:30**
Professor: Abdorrahim Bahrami **Page 1 of 14**

Name: ____

Student Number: ____

Signature ____

Notes:

- No calculators or other electronic devices are allowed.
- The answers for this test are to be written in the space provided. Use the last page and/or the backs of pages if extra space is required.
- The marks are allocated as follows:

Question	Marks	Marks Obtained
1	10 + 2	
2	10 + 8	
3	10	
Total	30 + 10	

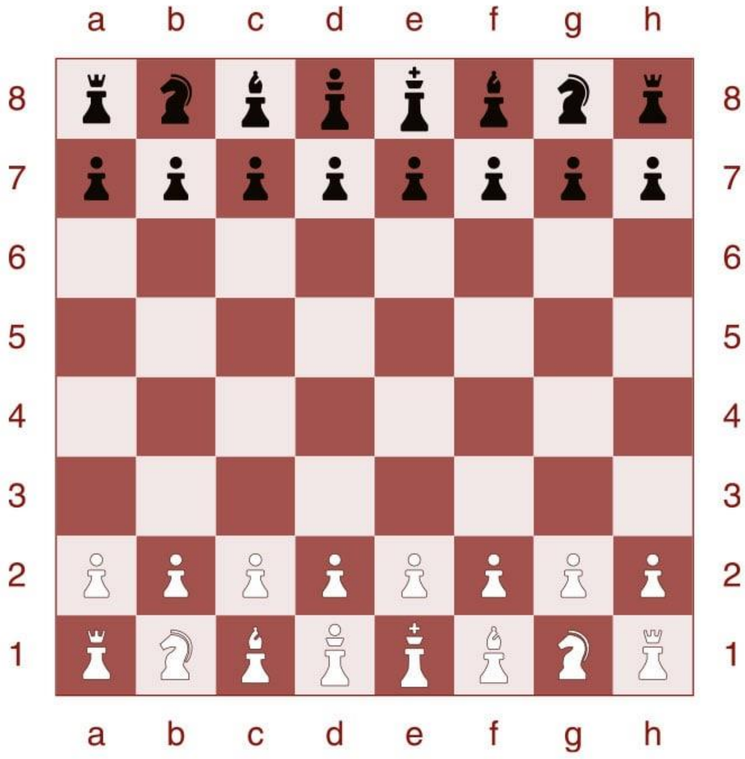
Q1- A chessboard is like the picture shown below with rows are shown by numbers 1 to 8 and columns are shown with characters 'a' to 'z'. Design a class called Piece, which represents pieces we have in chess. Each piece has a color, which can be white or black, and a current position. Then, you need to have a method called **move** that moves the piece to a new given position. The new position is given with an integer for the row and a character for the column. You need to inherit the following pieces from this class.

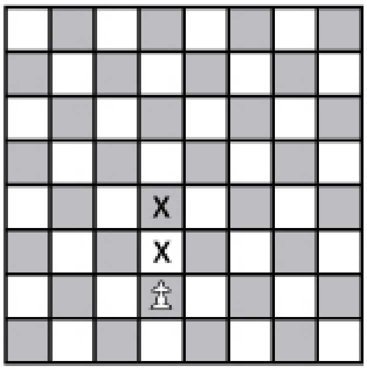
King, Queen, Rook, Knight, Bishop.

Pawn (Bonus mark), note that pawn can move one or two squares in the forward direction from its initial position but it can move one square only after in the forward direction. (Assume white pawns always start at row number 1 shown below, and black pawns start at row number 7 shown below)

You should check the validity of the new position, and if the new position is a valid move for the given piece, move the piece to that position. Have proper default and user constructors for classes.

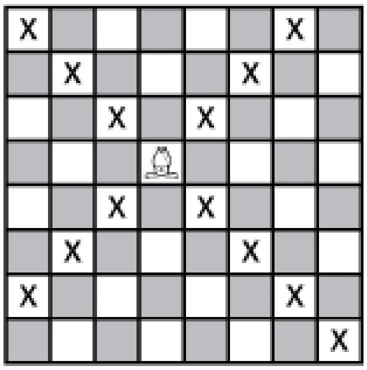
(10 + 3 marks)



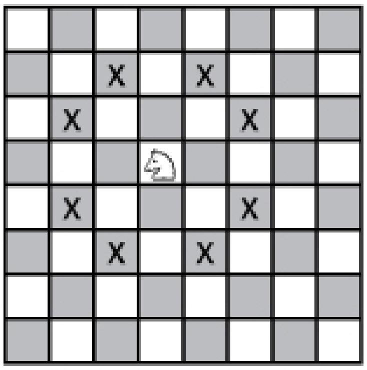


pawn

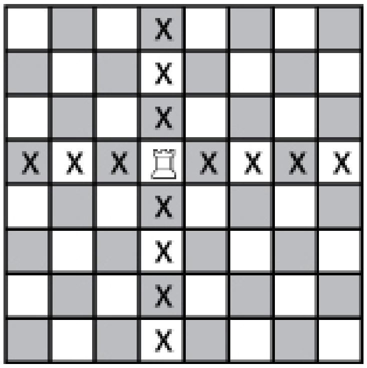
(can move 2 squares on 1st move only!)



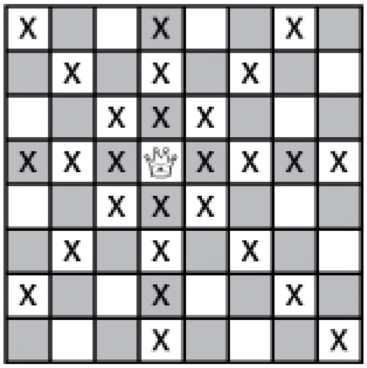
bishop



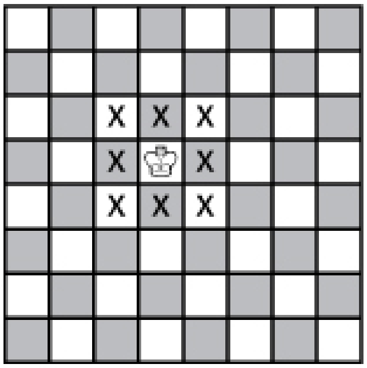
knight



rook



queen



king

Q2- Design a class to represent an image. An image is a 2-dimensional array of pixels. Each pixel has a non-negative integer value as intensity. Your class should have the following methods.

A default constructor that creates an image with a default size whose pixels have a default intensity.

A user constructor that receives the size of the image and an intensity and creates an image with the given size whose pixels have the given intensity.

operator -, that takes two images. If two given images have the same size, it subtracts the intensity of each pixel in the second image from the intensity of the pixel at the same position in the first image. Note that the result pixel is a pixel with a 0 intensity if a pixel at a position in the second image has a higher intensity. If the images do not have the same size, it just returns the first image.

operator +, that takes two images. If two given images have the same size, it adds the intensity of each pixel in the second image to the intensity of the pixel at the same position in the first image. If the images do not have the same size, it just returns the first image.

operator ++, which adds 1 to the intensity of each pixel in the image in both formats.

operator --, which subtracts 1 from the intensity of each pixel in the image in both formats. Note that when a pixel with intensity 0 is subtracted by 1, it still results in a pixel with intensity 0.

sub_image, it checks if the current image appears in part of the given image.

Then, inherit a class called Monochrome, which represents a gray-scale image. The intensity of each pixel is between 0 and 255 in monochrome images. For the + operator, you need to take the remainder by 256 to keep the intensity of pixels in range. (10 marks)

Bonus mark
Then inherit a class called Colored, which represents a colored image. Each pixel has three values, which represents red, green, and blue intensities in a colored image. All intensities are between 0 and 255. All operations above follow the same instruction separately for red, green, and blue.
You need to add three methods for this class. (8 marks)

GreenFilter, which keeps the green intensity and makes the other two 0s for each pixel.
BlueFilter, which keeps the blue intensity and makes the other two 0s for each pixel.
RedFilter, which keeps the red intensity and makes the other two 0s for each pixel.

Define the casting operator to cast a colored image to a monochrome image.
Hint: In a monochrome image, the intensities of red, green, and blue are the same.

Q3- Design a class of the concept of Fuzzy sets called FuzzySet. A fuzzy set is a set whose elements have a membership value between 0(exclusive) and 1(inclusive). **Members of the set can be of any types. Therefore, you need to use template.** A Fuzzy set of integers is used below just as an example. (8 marks)

Default constructor that creates an empty Fuzzy set.
Copy constructor
Destructor (If you need it)

intersection for the intersection of two fuzzy sets.
Note that if an element appears with the membership value $m1$ in the first fuzzy set and with the membership value $m2$ in the second fuzzy set, it will appear with the membership value of $\min(m1, m2)$ in their intersection.
$\{(5, 0.2), (10, 0.6), (7, 0.8), (1, 0.23)\} \cap \{(6, 0.27), (8, 0.92), (5, 0.14), (1, 0.72)\} = \{(5, 0.14), (1, 0.23)\}$

operator + for getting union of two fuzzy sets.
Note that if an element appears with the membership value $m1$ in the first fuzzy set and with the membership value $m2$ in the second fuzzy set, it will appear with the membership value of $\max(m1, m2)$ in their union.
$\{(5, 0.2), (10, 0.6), (7, 0.8), (1, 0.23)\} + \{(6, 0.27), (8, 0.92), (5, 0.14), (1, 0.72)\} =$
$\{(5, 0.2), (10, 0.6), (7,0.8), (1, 0.72), (6,0.27), (8,0.92)\}$
Note that when an element is not in a fuzzy set, it means its membership value is 0.

operator - for the difference of two fuzzy sets.
Note that if an element appears with the membership value $m1$ in the first fuzzy set and with the membership value $m2$ in the second fuzzy set, it will appear with the membership value of ~~min~~$(m1 - m2, 0)$ in their difference.
$\{(5, 0.2), (10, 0.6), (7, 0.8), (1, 0.23)\} - \{(6, 0.27), (8, 0.92), (5, 0.14), (1, 0.72)\} =$
$\{(5, 0.06), (10, 0.6), (7,0.8)\}$

operator equality for checking equality of two fuzzy sets.
Two fuzzy sets are equal if they have the same members and each member has the same membership value.

operator << for printing a fuzzy set in the following format,
$\{(5, 0.06), (10, 0.6), (7,0.8)\}$