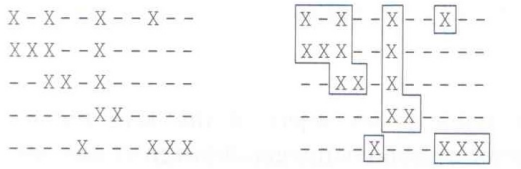
Assignment 1

1 Assignment Policy

- Weight: this assignment is worth 20% of your final grade.
- Grading: out of 35 points, as explained below.
- Due Date: Oct 24,2015 11:50PM
- Late Submission policy: 10% penalty up to five days after the due date. NO ASSIGNMENT WILL BE ACCEPTED 5 days after the deadline.
- Deliverables: One Microsoft Word document and one Visual Studio project (as explanined below) zipped in a single zip file using this naming convention: if your name is John Smith, the file name must be JohnSmithA1.zip. The required file must be posted on Balckboard in the Assignment 1 folder created for this purpose.
- The submitted work must be individual. All submissions will be compared using software and if two submissions are similar, they will be punished according to George Brown Academic Honesty Policy.

2 Recongizing blobs in an image

An image is just a matrix of pixels. For simplicity, we will use 1 bit to represent 1 pixel. If we place rows of the matrix one after another, we get an array. Therefore we assume an image can be stored in a BitArray variable where BitArray is the class defined in week 4. For your convenience, the relevant code will also be attached to this assignment.
In order to work in text mode, without the complication of graphics, we will represent an image as a matrix where the pixels that consitute blobs in the image are repsented by letter "X". The pixels that do not belong to any of the blobs, will be repesented by dashes "-".
An example of such representation is below:



By definition, for our exercise, a *blob* is a group contiguous letters "X". Two "X"-s in a grid are considered contiguous if they are beside each other horizontally or vertically. For example, the image above contains five blobs, as indicated in the correposnding grid to the right where blobs are outlined.

(a) (2 Points) Give a recursive definition of a blob.

(b) (1 Point) We will use a BitArray variable to represent a grid. Note BitArrays are one dimensional and our grid is two dimensional. What would be the formulas that identify the index of the bit representing the pixel with coordinates (row, col) on the grid?.

(c) (1 Point) Place the results of the items (a) and (b) in a MS Word file named JohnSmithA1 using same name convention as explained above.

(c1) (1 Point) Create a Visual Studio project using same naming convention. **If you use a project name different from the specified name, 5 marks will be deducted**. Add the given BitArray.h to your project and create the main.cpp the usual way.

(d) (4 points) Create a function called `genGrid` with the following signature:
`BitArray genGrid(unsigned int rows, unsigned int cols, unsigned int percentage)`
that generates a grid, where each pixel belongs to a blob based on a random number generated in the range 1 to 100: if the random number is less than the percentage, the pixel belongs to some blob (that is you set the corresponding bit to 1); otherwise the pixel does not belong to the blob (that is you set the corresponding bit to 0).

(e) (4 Points) Create a function `gridView` with the following signature:
`string gridView(BitArray &grid, unsigned int rows, unsigned int cols)`
that produces a string representation of the `grid` in a form similar to that shown on the image above.

(f) (8 Points) In order to count the blobs, you will create a function with the following signature:

```
int blobCount(BitArray &grid, BitArray visited,
 unsigned int rows, unsigned int cols)
```

where `grid` is the representation of the image, and `visited` keeps track of the fact if your counting function has ever visited a particular element of the grid. Note if you mark a pixel as visited, you want to check the neighbor pixels and amrk them as well if they belong to the same blob.

(g) (10 Points) The marking of pixels belonging to blobs must be done using a recursive function with the following signature:
`void markBlob(BitArray &grid, BitArray &visited, unsinged int rows, unsigned int cols,unsigned int row, unsigned int col)`
where `row, col` is the particular pixel that is being visited at this time.

(h) (4 Points) Your `main()` program must prompt the user for the number of rows and number of columns of a grid, and also for the percentage of the blobs, and then generate a grid, count the blobs, and show the grid and the count as indicated in the following screenshot: