

Name: _____

Problem	Points	Score
1	50	
2	50	
Total	100	

Notes:

- (1) Please follow the instructions in this exam for how to submit your work.
- (2) You can use all the web resources at your disposal except other human beings (and talking to someone via a chat line counts as an interaction with a human being 😊)

The first step in this exam is to create a workspace in the following directory:

`/data/courses/ece_3822/current/exams/exam_02`

Your directory should be your last name all lowercase (e.g, “picone”). Set the permissions using “`chmod u+rwx,g-rwx,o-rwx <lastname>`” so only you have read and write permission to this directory. Create two subdirectories within this directory: `p01` and `p02`. You will use these for problems 1 and 2 respectively. Put ALL your code in these directories.

Problem No. 1: This file:

`/data/courses/ece_3822/2019_spring/exams/exam_02/picone/p1.dat`

contains short integers that range from `[-11,10]`:

```
nedc_999_[1]: od -s p1.dat
0000000  -11  -10  -9   -8   -7   -6   -5   -4
0000020   -3   -2  -1    0    1    2    3    4
0000040    5    6    7    8    9   10
0000054
```

Create a C++ class that implements a circular buffer. Your driver program should support this interface:

`p01.exe <filename> N`

where `N` is the buffer size. Your program reads the file sample by sample, adds the sample to the circular buffer and prints the contents of the buffer after each read. For example, for `N = 4`, your output will look like this:

```
0 -11 [-11  0  0  0]
1 -10 [-10 -11  0  0]
2 -9  [-9 -10 -11  0]
3 -8  [-8 -9 -10 -11]
4 -7  [-7 -8 -9 -10]
...
```

Your circular buffer should not do any block memory moves or shifts. You need to use a single buffer `N` samples long and use pointers to control the location of the data (and to print the data).

You must write your own code. Do not try to copy code from the Internet. Your class should have a function to construct a buffer, to print the contents in the format above, and to add a sample to the buffer.

Problem No. 2: Write a class that encapsulates a pair of numbers: `(x,y)`. Implement an overloaded version of the addition operator. Write a main program that implements this code:

```
MyPair a(3,4);
MyPair b(5,6);
MyPair c;
c = a + b;
float c_x = c.getx();
float c_y = c.gety();
fprintf(stdout, "the sum of this pair is: %f, %f\n", c_x, c_y);
```

The addition operator, `+`, should add the first elements together, then add the second elements together, and then save those in a new object.

Hint: To implement this you might need to implement a few additional member functions.