Homework # 9 - Piecewise Interpolation with Cubic Splines

Objective: To write a ***cubic spline interpolation module*** interpolation for arbitrary sets of data points in C.

ap_hw9files:

interp.c, interp.h	Interpolation code frame work
data.txt	Simple data to use to generate the spline coefficients. The first line always contains the derivatives at the endpoints (y_o^0, y_n^0) Each remaining line contains one data point (x_i, y_i). The final test will use much longer data.
verboseDataOutput.txt	A data dump of my verbose implementation to help verify your implementation

Requirements:

The program **hw9** will take command line arguments to perform different tasks as follows:

- If called with no arguments (e.g, **hw9**) a friendly "usage" message should be printed. The friendly message should include a brief description of the program, the syntax and the meaning of key options.
- Standard syntax behavior **hw9 -cl -p[oints] data.txt <-v[erb[ose]]>** (-v, -verb & -p allowed)
- Your implementation should be able to handle an arbitrary number of data points (you do not know how many in advance). You must reuse your DynamicArray code (global variables are **not allowed)**. There is a HW9 section in your DynamicArrays.h code, use the **-DHW9** compile option to enable your changes.
- The program should output to **stdio** in the following "tabular form": X_j x_{j+1} d_j c_j b_j a_j

```
X0, X1, d, c, b, a   N= 10
 0.0000000   0.3141593  14.9167590  -11.0524281    2.0000000    0.0000000
 0.3141593   0.6283185  -4.1670526    3.0062863   -0.5277700    0.0000008
 0.6283185   0.9424778   4.1223494   -0.9210683    0.1273205    0.0017007
…etc for all data points….
```

Makefile:

You must provide a quality Makefile with the following targets:

"all" -should make **hw9**
"cl " - should run **hw9**, in the corresponding mode with **data.txt,** redirecting output to a file of the form: **out_cl.txt**.
"mem" - should run **hw9** in **clamped** mode using valgrind redirected to **mem.txt.**
help, clean - should do the normal things

Your code must compile incrementally e.g. each .C file generate a corresponding .o file and then links the .o files to produce the binary.

Testing and Analysis:

Prepare an analysis file, **analysis.txt** that included a short explanation of how you verified that the calculations were correct and any other relevant aspects of your approach. It should at least include the following tests:

i. Verify that the program works correctly for different number of command line arguments.
ii. Ensure that the module works correctly for arbitrary number of data points (e.g. no hard coding fix array sizes). Use redirection to save the parameters of the splines to a text file called **out_cl.txt.**

Once all of the above tests are completed create and submit a the tar file **lastName_hw9.tar** (lastName is your last name) with all the relevant files.

Grading:

1. (80 points) Module Implementation
   a. (30 points) Spline generation algorithm generates the correct results.
   b. (20 points) Program handles any number of data points.
   c. (20 points) Program has no memory leaks.
   d. (10 points) **makefile** works properly for compiling and testing.

2. (20 points) Analysis
   a. (20 points) Clear and concise **explanation** of implementation and validation process.