```
Guideline for Warmup #1
=======================

Total points: 100

+---------------------+
| General Guidelines: |
+---------------------+

1) The points below are maximum points.  Partial credits may be given.

2) Do the "plus points" first, then do the "minus points".

3) Please read the student's README first to see if you need to do something
   different to get their code to work and what the known problems are.

4) The scripts below are for csh/tcsh.  If you use bash, please modify
   them where necessary.  Or, you can do "tcsh" and then run the scripts.

5) The grading is meant to be harsh!  So, if running the "diff" command
   suppose to produce no output, but the student's code produces some
   output, take points off accordingly.  Similarly, if running the "diff"
   command suppose to produce some output, but the student's code
   produces no output, take points off accordingly.

6) The minimum deduction for a mistake of any kind is 0.5 pt (unless you have
   lost all the points allocated for a particular test/subtest) even if all
   are caused by the same programming bug.

7) The posted grading guidelines is not perfect.  We will try our best to
   stick to the grading guidelines.  Sometimes, after we started grading
   some of the submissions, we realize that we missed something (most often
   in "minus points").  The bottomline is that we reserve the right to add
   or change things in the grading guidelines as we see fit.

+----------------+
| Do this first: |
+----------------+

    if (-f my402list.h) then
        mv my402list.h my402list.h.submitted
    endif
    if (-f cs402.h) then
        mv cs402.h cs402.h.submitted
    endif
    cp ~csci570b/public/cs402/warmup1/cs402.h .
    cp ~csci570b/public/cs402/warmup1/my402list.h .
    make warmup1

+--------------+
| Plus points: |
+--------------+

(A) Doubly-linked Circular List : +40 points
                set srcdir=~csci570b/public/cs402/warmup1

                #
                # If using C (well, you have to use C):
                #
                        /bin/rm -rf grading_$$
                        mkdir grading_$$
                        cd grading_$$
                        cp ../my402list.c .
                        cp $srcdir/cs402.h .
```

```
                        cp $srcdir/my402list.h .
                        cp $srcdir/listtest.c .
                        cp $srcdir/Makefile .
                        make

                        set seeds = ( 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 )

                        #
                        # for the following commands, each correct behavior gets 2 point
                        # gets 2 points if "./listtest" command produces NOTHING
                        # gets 0 point if "./listtest" command produces ANY output
                        #
                        foreach f (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
                            echo "===> test_$f"
                            ./listtest -seed=$seeds[$f]
                        end
                        cd ..
                #
                # Clean up temporary directory
                #
                /bin/rm -rf grading_$$

 (B) Sort : +60 points
                set srcdir=~csci570b/public/cs402/warmup1

                #
                # (B1)
                # for the following commands, each case is worth 2 points
                # gets 2 points if "diff" command produces NOTHING
                # gets 1 point if "diff" command produces SOME output showing
                #     very minor formatting errors (i.e., extra or missing space/tab/cr
 characters)
                # gets 0 point if "diff" command produce SOME output showing
                #     incorrect information in the output
                #
                /bin/rm -f f?.sort f??.sort
                foreach f (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14)
                    echo "===> $srcdir/f$f"
                    ./warmup1 sort $srcdir/f$f > f$f.sort
                    diff $srcdir/f$f.sort f$f.sort
                end
                /bin/rm -f f?.sort f??.sort

                #
                # (B2)
                # for the following commands, each case is worth 2 points
                # gets 2 points if "diff" command produces NOTHING
                # gets 1 point if "diff" command produces SOME output showing
                #     only invisible formatting errors
                # gets 0 point if "diff" command produce SOME output showing
                #     incorrect information in the output
                #
                /bin/rm -f f??.sort
                foreach f (15 16 17 18 19 20 21 22 23 24 25 26 27 28 29)
                    echo "===> $srcdir/f$f"
                    cat $srcdir/f$f | ./warmup1 sort > f$f.sort
                    diff $srcdir/f$f.sort f$f.sort
                end
                /bin/rm -f f??.sort

 +---------------+
 | Minus points: |
 +---------------+

 Missing/incomplete required section(s) in README file : -20 points (even if lots of comments
```

```
in code)

Submitted binary file : -2 points

Cannot compile      : -5 to -10, depending on effort to make it work
                      if cannot generate executable, at least 5 points
                          will be taken off, up to 10 points off if it
                          takes a lot to make it work

Compiler warnings   : -1 to -3
                      if using gcc/g++, make sure -Wall flag is used
                      please take 1 point off for 1-4 warnings,
                      2 points off for 5-8 warnings, 3 points for more

"make clean" : -2 points if cannot delete all binary files created during compilation
               (.o files, .gch files, executable files, etc.)

Segmentation faults : -5 to -10 points
                # if you have seen that the student's program gets segmentation
                #         faults or bus errors any time during your testing of
                #         the code, deduct 5 points, any additional occurrance,
                #         deduct 0.5 pt each

Program never terminates : -5 to -10 points
                # if you have seen that the student's program won't terminates any
                #         time during your testing of the code, deduct 5 points,
                #         any additional occurrance, deduct 0.5 pt each

Separate compilation : -10 points
                if the executable is compiled with a single line, deduct
                    all 10 points
                if the executable is a single module, but generates warmup1.o and
                    then link to create the executable in line one, deduct 5
                    points
                if a lot of the code are in .h files, deduct all 10 points

Malformed input : -2 point each for each not handling properly (should print
                    a reasonable error message and quit, just saying that
                    there's something wrong with the input is not reasonable,
                    you must give a line number and a field name if applicable)
                set srcdir=~csci570b/public/cs402/warmup1

                cat $srcdir/f101 | ./warmup1 sort
                cat $srcdir/f102 | ./warmup1 sort
                cat $srcdir/f103 | ./warmup1 sort
                cat $srcdir/f104 | ./warmup1 sort
                cat $srcdir/f105 | ./warmup1 sort
                cat $srcdir/f106 | ./warmup1 sort
                cat $srcdir/f107 | ./warmup1 sort

Too slow : -5 points for running way too slow
                # if correct results are produced but just run too slowly,
                #     take points off according the slowest run of the
                #     program (i.e., worst case)
                # if it takes 15 seconds or more to run, take 1 point off
                # if it takes 30 seconds or more to run, take 2 points off
                # if it takes 45 seconds or more to run, take 3 points off
                # if it takes 1 minute or more to run, take 4 points off
                # if it takes 2 minute or more to run, take 5 points off

Bad commandline or command : -1 point each for each not handling properly
                # need some sort of error output to indicate that the command
                #     cannot be completed, it does not have to be exactly the
                #     error indicated below, but it MUST BE something reasonable
                #     to inform the user about the error condition
```

```
                # if a commandline is malformed, MUST in addition give "usage"
                #     information as described in the spec
                # please note that commandline parsing is separate from
                #     other functionalities, so even though a student has
                #     declared that certain part of the assignments is not
                #     imlemented, commandling parsing still needs to be done
                #     for those commands
                set srcdir=~csci570b/public/cs402/warmup1

                ./warmup1
                    (malformed command)
                ./warmup1 -y sort
                    (malformed command)
                ./warmup1 xyz
                    (malformed command)
                ./warmup1 abc def ghi
                    (malformed command)
                ./warmup1 ksjdfjwiejofjasdfjowkejokjaosijfioejfsiejriwjeirjwier
                    (malformed command)

                ./warmup1 sort -x
                    (malformed command or input file "-x" does not exist)
                ./warmup1 sort /usr/bin/xyzz
                    (input file /usr/bin/xyzz does not exist)
                ./warmup1 sort /etc/sysidcfg
                    (input file /etc/sysidcfg cannot be opened - access denies)
                ./warmup1 sort /etc/inet/secret/xyzz
                    (input file /etc/inet/secret/xyzz cannot be opened - access denies)
                ./warmup1 sort /etc
                    (input file /etc is a directory)
                ./warmup1 sort /etc/motd.all
                    (input file is not in the right format)
                ./warmup1 sort ~/.login
                    (input file is not in the right format)

 Did not use My402List and My402ListElem to implement "sort" in (B) : -30 points
                #
                # inspect student's code to see if another list implementation
                # is used or an array is used to sort the list (e.g., using qsort())
                #
```