Assignment 01
PARSING REAL CONSTANTS

DS286.Aug16 Data Structures and Programming

August 20, 2016

Submission is due on or before **Sunday, August 28, 2016, 11:59pm IST**.
The assignment carries **50 points**, which is 5% of the course weightage.

1 Question

You are given a *List* data structure which is implemented using *Arrays* in C++. It has the ability to store *double precision real numbers* in it.

Complete the code in the given `main.cpp` file so that when the program runs, the following behavior is exhibited by the program.

The program accepts a *string* as a command line argument. The string may contain *any printable characters* other than a white space [1]. Of these, characters {`a`,`d`,`p`,`A`,`D`,`P`} will be called *action characters*. Let i be a position in the string where an action character is found, and let k be the first position $k > i$ where another action character or a `NULL` (`\0`) is found. Your program should take the following action for each action character found in the input.

1. **Append:** When an action character `a` or `A` is found, then (1) check if the substring between $i+1$ and $k-1$ forms a *valid real number*, and (2) if so *append the real number to the List.*

2. **Delete:** When an action character `d` or `D` is found, then (1) check if the substring between $i+1$ and $k-1$ forms a *valid integer*, and (2) if so *delete the item in that integer position from the List.*

3. **Print:** When an action character `p` or `P` is found, *print the current items in the list.*

If the parsing of the real or integer number fails for append or delete, that particular action should not be performed. If the integer does not correspond to a valid position in the List for delete, that action should not be taken. Characters that appear before the first action character can be ignored as well.

[1] `http://en.cppreference.com/w/cpp/string/byte/isprint`

2 Sample Inputs

The following test cases illustrate the correct behavior of the program, assuming the output executable file is `listdemo.out`.

```
./listdemo.out a1a2a3a4pd0p
[1,2,3,4] [2,3,4]

./listdemo.out a1a2a3xxa4pd-12.3pa7
[1,2,4] [1,2,4]

./listdemo.out a-1.234e-10a2a5a1.e10d1d1p
[-1.234e-10,1.e10]

./listdemo.out a157a2.0a3.0a0.0a.1a1.0a.1e10a1e23p
[157,2.0,3.0,0.0,0.1,1.0,0.1e10,1e23]

./listdemo.out xxxa-1.234e-10a1.e10a1.234e10a1.234e-10p
[-1.234e-10,1.e10,1.234e10,1.234e-10]

./listdemo.out xxx+1.234e+10a+1.234g+10a+1.234e10x+10d-2p
[]
```

The format in which the numbers are printed is just for illustration. For example, 1.0 might be printed as 1.000 or 1e00 depending upon the print function. The function which prints the list is already written and you *must not modify it* from its original format.

The test cases used to evaluate your submission will be different from the above examples. Any legal test case which follows the problem definition can be used in final evaluation. Your program *should not fail* for any valid input, and should provide the correct answer.

3 Submission Instruction

Please follow these instructions carefully. We use automated scripts for evaluation. So a failure to follow these instructions will mean that your submission will not be evaluated.

- Write your program by completing the skeleton code that is provided. Do not modify the original skeleton itself. *Just add extra code to complete the assignment in the space indicated.*

- Put all your files including source file, executable file and *Makefiles*, in a single folder whose name is determined as follows. My full name is "Prateeksha Varshney" where "Prateeksha" is my first name, so the folder name should be `01Prateeksha` for my submission. Please note the capitalization of first letter of the first name.

- This folder should be compressed using the `tar` program as follows:
  `tar -cvf 01Prateeksha.tar 01Prateeksha/`

  Note: Any other compression format *will not be accepted* and will be treated as no submission. Its your responsibility to check if the file can be properly uncompressed and all files inside are intact.

- Send a separate mail to the TA's email address `prateeksha@grads.cds.iisc.ac.in` with the subject line `DS256.Aug16.A01`. Do not write anything more or less in the subject line. Do add any text in the body. Do not send the assignment as a reply-email to any other mail.

- **Only one submission will be accepted**. If multiple emails or files are received, **only the first one** will be taken as the submission. Only the submission received **before the deadline** will be accepted.

- Use only the C++ language for completing this assignment. Make sure the code compiles and executes correctly on the head node of the `turing.cds.iisc.ac.in` server using `g++` command. The code will be compiled and tested on this machine.

4 Ethics

You should not get assistance from other students or external sources in directly solving the assignment. Getting help on generic C++ and data structures concepts, e.g., on using lists, strings, libraries, compilation, etc. is accepted. You are encouraged to post questions to the course mailing list so that the TA, instructors or other students can respond. This also ensures you do not have an unfair advantage/disadvantage over other students. If you have received assistance from other sources, send a *separate email to the Instructor and the TA* disclosing the external sources and type of support received.

By making a submission, you are asserting that all code that you submit was designed and developed by you. Do NOT copy and paste code from anyone else! All code will be verified using a plagiarism checker, and *penalties will be imposed* if plagiarism is found from unattributed sources.