|  |  |
| --- | --- |
| Student id: | 100357736 |
| Student name: | Seung Yeon Lee |

Part I – MCQ (10 questions, 1 mark each) – administered using D2L [25%]

Part II – Programming [75%]

* Hands-on
* Upload all files to D2L (.h, .cpp, Makefile)
* To earn marks, your code must compile and run (working)
* Do not hardcode the output
* 1% deduction for unnecessary codes
* 1% deduction for requirements not followed
* Provide proof that your features work

Part II – Programming [30 marks]

Data file: pokemon.csv

The 1st row is the header

The 2nd to Nth row represents the data

Write a C++ application with the following features

1. loadFile (design and create your own parameters)
   * reads pokemon.csv
     1. store the header in an array of string (e.g. string header[])
     2. store the data in a double LinkedList
        1. create a class that would represent each row. It should implement
           1. constructor
           2. destructor
           3. setters and getters
           4. declare the variables as private

string name

int total

int hp

int atack

int defence

int sp\_attack

int sp\_defence

int speed

* + - * 1. operator<<

display the name, total, hp, attack, defence, sp\_attack, sp\_defence, speed

provide proper label to describe what the data represents



should be properly formatted

* + - 1. Separate the declaration (.h file) and implementation (.cpp file)
      2. Filenames: Pokemon\_yourStudentid.h and Pokemon\_yourStudentid.cpp
    1. The double LinkedList
       1. Must implement
          1. template
          2. constructor
          3. destructor
          4. push(T data)

adds the data at the front of list

* + - * 1. append(T data)

adds the data at the end of the list

* + - * 1. insert(int index, T data)

adds the data at the given location (index)

* + - * 1. clear

deletes all the nodes

sets size to 0

* + - * 1. getSize

returns the number of nodes

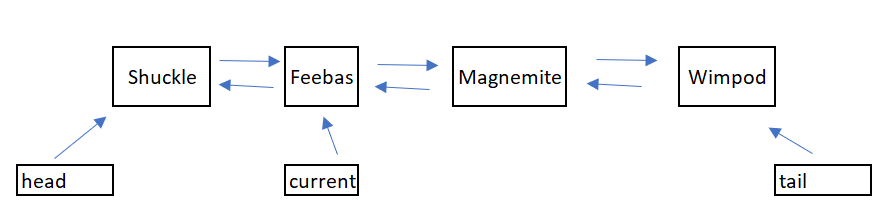
* + - * 1. T get(int index)

returns the element from a specified location

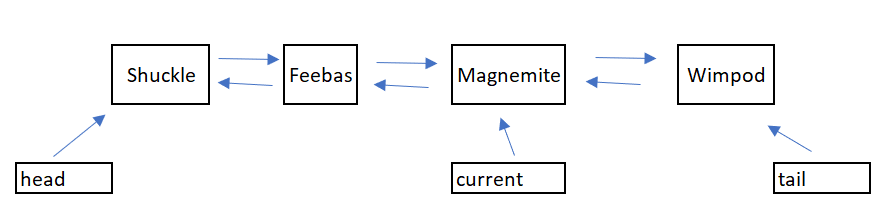
* + - * 1. T moveNext

Returns the next element

Example



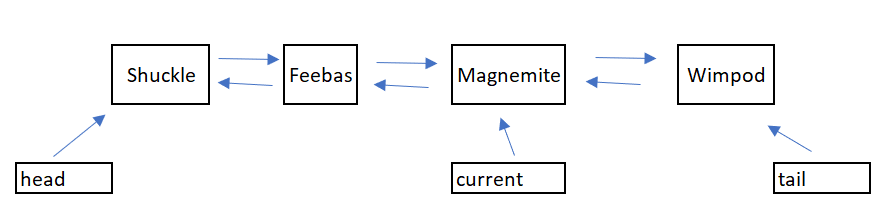
moveNext would move current to the next node and returns Magnemite (and the other data like hp, speed, etc..)



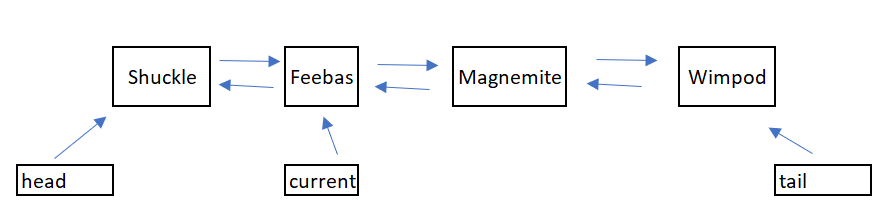
* + - * 1. T movePrev

Returns the previous element

Example



movePrev would move current to the previous node and returns Feebas (and the other data like hp, speed, etc..)



1. Menu

1. Display all

2. Display one record at a time

3. Contenders

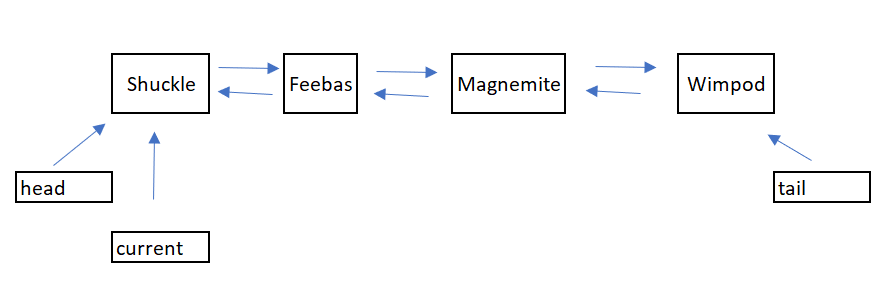
4. Exit

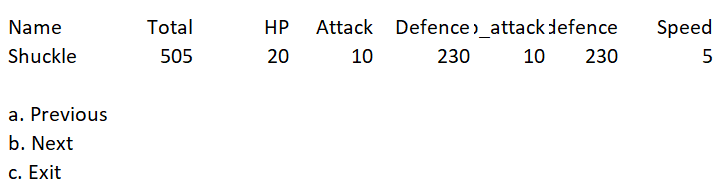
**Display all**

* Display all the data (e.g. name, hp, attack, …) to the console. It should be properly formatted for easy reading and sorted in descending order by name using QuickSort or Insertion Sort. Indicate in your code which algorithm you are using.
* QuickSort ‘s or Insertion Sort’s implementation should be using template

**Display one record at a time**

(illustration is just for example purpose)





If the user selects previous and no node exists, provide a friendly message to the user.

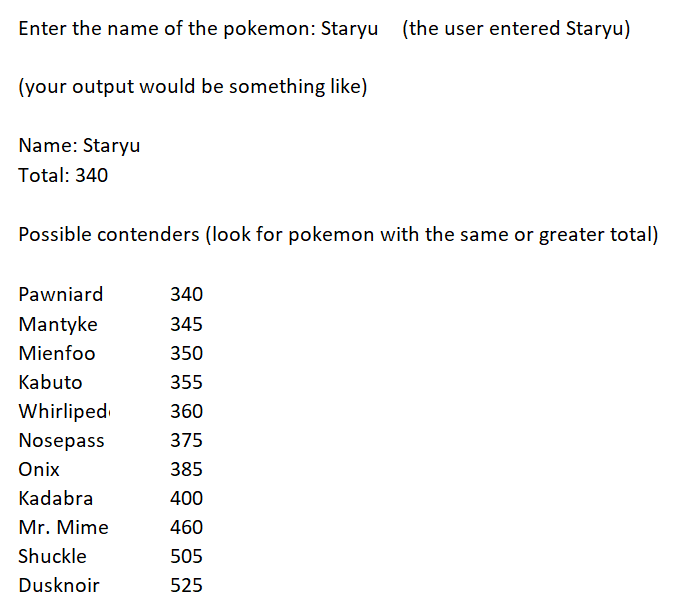
If the user selects next or previous, move the “current” appropriately if a node exists then display the data

In the example give, the “current” would point to Feebas if the user selects “Next”

You must use the operator<< that you implemented in Pokemon class

Contenders

* Ask the user for a pokemon’s name then your application provides the possible contenders based on total
* Example (do not display the statements in enclosed in parenthesis)



Things to submit

1. .h and .cpp files
2. Makefile

Put all your files in a zip, FinalExam\_yourstudentid.zip

|  |  |  |
| --- | --- | --- |
|  |  | marks |
| menu |  | 0.75 |
| loadFile |  | 15 |
| Display all |  | 4.5 |
| Display one record at a time |  | 4.5 |
| Contenders |  | 4.5 |
| Makefile |  | 0.75 |