

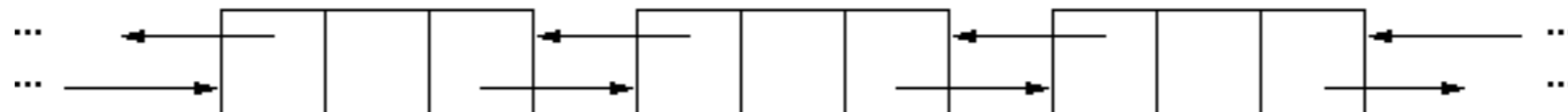
[PA #2] UniversalList

Submission deadline: ~ 11/12 11:59 AM

UniversalList?

- Doubly linked list (unvlist & node classes)
- List of elements can have various data types (#9)
 - bool, int, char, short, long, long long, float, double, and string
- Sequential access only
- Supports various methods (#13)

A Doubly-Linked List



Support Methods I

- constructor ()
 - Creates an empty doubly linked list
- constructor (const int* arr, int array_length)
 - Creates a list with elements in integer array *arr*
- destructor()
 - Free the allocated list

Support Methods II

- **operator+**
 - Concatenates the two given lists A and B
 - e.g. list C = list A + list B
- **operator==**
 - Check whether list A and B have same elements or not
 - e.g. if (list A == list B) return true / false

Support Methods III

- `set (int n, const value_type& val)`
 - Updates the element of index n with the value *val*
- `get (int n)`
 - Returns the element of index n

Support Methods IV

- `insert (int n, const value_type& val)`
 - Inserts & returns value *val* at the index *n* of the list
- `erase (int n)`
 - Removes & returns the element at the index *n* of the list

Support Methods V

- size (void)
 - Returns the length of the list
- pop_back (void)
 - Removes the last element of the list
- pop_front (void)
 - Removes the first element of the list

Support Methods VI

- `push_back (const value_type& val)`
 - Inserts the value *val* at the end of the list
- `push_front (const value_type& val)`
 - Inserts the value *val* at the beginning of the list

Additional Information

- Skeleton files are given
 - `main.cpp` <- You modify this file
- ~~Given *Makefile* has following rules~~
 - ~~`$make:` build an executable file~~
 - ~~`$make clean:` clean generated files from make command~~
- Your program will be graded using *main()* and *main*()*
 - *main*()*: more complicated test cases (not open to the public)