

Homework #1: Circular Doubly Linked List

- Deadline: Apr. 17, 11:59 PM
- Recommended: Use Visual Studio 2022 or Visual Studio 2019
- Submission Requirements: 김철수_20251111.zip
 1. Report: PDF format (2~3 pages, plus appendices for code & comments only)
 2. Source Code: Files in .cpp and .h format
 3. Executable: Compiled .exe file
- Grading Criteria:
 - Code / Executable: 50%
 - Report: 50%

```
#include <iostream>
#include "CDList.h"
```

```
int main() {
    CDList playList;

    std::cout << "add(SA)" << std::endl;
    playList.add("SA");
    std::cout << playList << std::endl;

    std::cout << "add(LF)" << std::endl;
    playList.add("LF");
    std::cout << playList << std::endl;

    std::cout << "add(JT)" << std::endl;
    playList.add("JT");
    std::cout << playList << std::endl;

    std::cout << "forward()" << std::endl;
    playList.forward();
    std::cout << playList << std::endl;

    std::cout << "forward()" << std::endl;
    playList.forward();
    std::cout << playList << std::endl;
```

main.cpp

```
add(SA)
Forward hopping: SA*
Backward hopping: SA*

add(LF)
Forward hopping: LF->SA*
Backward hopping: LF->SA*

add(JT)
Forward hopping: JT->LF->SA*
Backward hopping: LF->JT->SA*

forward()
Forward hopping: LF->SA->JT*
Backward hopping: SA->LF->JT*

forward()
Forward hopping: SA->JT->LF*
Backward hopping: JT->SA->LF*
```

```

std::cout << "remove()" << std::endl;
playlist.remove();
std::cout << playlist << std::endl;

std::cout << "add(DI)" << std::endl;
playlist.add("DI");
std::cout << playlist << std::endl;

std::cout << "backward()" << std::endl;
playlist.backward();
std::cout << playlist << std::endl;

std::cout << "backward()" << std::endl;
playlist.backward();
std::cout << playlist << std::endl;

std::cout << "remove()" << std::endl;
playlist.remove();
std::cout << playlist << std::endl;

std::cout << "remove()" << std::endl;
playlist.remove();
std::cout << playlist << std::endl;

```

```

remove()
Forward hopping: JT->LF*
Backward hopping: JT->LF*

add(DI)
Forward hopping: DI->JT->LF*
Backward hopping: JT->DI->LF*

backward()
Forward hopping: LF->DI->JT*
Backward hopping: DI->LF->JT*

backward()
Forward hopping: JT->LF->DI*
Backward hopping: LF->JT->DI*

remove()
Forward hopping: LF->DI*
Backward hopping: LF->DI*

remove()
Forward hopping: DI*
Backward hopping: DI*

```

```
std::cout << "remove()" << std::endl;
playList.remove();
std::cout << playList << std::endl;

std::cout << "remove()" << std::endl;
playList.remove();
std::cout << playList << std::endl;

return EXIT_SUCCESS;
}
```

```
remove()
ERROR: Cannot print. The list is empty.

remove()
ERROR: Cannot remove from an empty list.
ERROR: Cannot print. The list is empty.
```

CDList.h

```
✓ #ifndef CDLIST_H
  | #define CDLIST_H
✓ #include <iostream>
  | #include <string>
  |
  | using std::string;
  | using std::ostream;
  |
  | typedef string Elem;
  |
  | .....
  |
  | .....
  |
  | L #endif
```

CDList.cpp

```
#include "CDList.h"
```

```
CDList::CDList() : cursor(nullptr) {}
```

.....

.....

Additional Result

- e.g.)

```
Enter any string:
delft
add(user_input)
Forward hopping: delft*
Backward hopping: delft*

Enter any string:
boston
add(user_input)
Forward hopping: boston->delft*
Backward hopping: boston->delft*

Enter any string:
LA
add(user_input)
Forward hopping: LA->boston->delft*
Backward hopping: boston->LA->delft*

Enter any string:
exit
```