

Lab 3

Part 1:

Description:

Walkthrough output analysis of a regular and sentinel doubly linked list.

```
#include "lab3.h"
#include <iostream>

int main(void){
    DList<int> regular;
    Sentinel<int> sentinel;

    std::cout << "testing regular doubly linked list" << std::endl;

    for(int i=0;i<3;i++){
        regular.push_front(i);
        regular.print();
    }
    regular.reversePrint();
}
```

Outputs:

regular.print():

0
1 0
2 1 0

regular.reversePrint():

0 1 2

```
for(int i=3;i<6;i++){
    regular.push_back(i);
    regular.print();
}

regular.reversePrint();
```

Output:

regular.print():

2 1 0 3
2 1 0 3 4
2 1 0 3 4 5

regular.reversePrint():

5 4 3 0 1 2

```
for(int i=0;i<7;i++){
    if(i%2==0){
        regular.pop_front();
    }
    else{
        regular.pop_back();
    }
    regular.print();
}
regular.reversePrint();
```

Output:

regular.print():

1 0 3 4 5

1 0 3 4

0 3 4

0 3

3

nothing

empty list

regular.reversePrint():

empty list

```
for(int i=0;i<3;i++){
    regular.push_back(i);
    regular.print();
}
regular.reversePrint();
```

Output:

regular.print():

0

0 1

0 1 2

regular.reversePrint():

2 1 0

```
for(int i=3;i<6;i++){
    regular.push_front(i);
    regular.print();
}
regular.reversePrint();
```

Output:

regular.print():

3 0 1 2

4 3 0 1 2

5 4 3 0 1 2

regular.reversePrint():

2 1 0 3 4 5

```
for(int i=0;i<7;i++){
    if(i%2){
        regular.pop_front();
    }
    else{
        regular.pop_back();
    }
    regular.print();
}
regular.reversePrint();
```

Output:

regular.print():

4 3 0 1 2

4 3 0 1

3 0 1

3 0

0

nothing

empty list

regular.reversePrint():

empty list

```
for(int i=0;i<3;i++){
    regular.push_front(i);
    regular.print();
}
regular.reversePrint();
```

Output:

regular.print():

0
1 0
2 1 0

regular.reversePrint():

0 1 2

```
std::cout << "testing sentinel list" << std::endl;

for(int i=0;i<3;i++){
    sentinel.push_front(i);
    sentinel.print();
}
sentinel.reversePrint();
```

Output:

sentinel.print():

0
1 0
2 1 0

sentinel.reversePrint():

0 1 2

```
for(int i=3;i<6;i++){
    sentinel.push_back(i);
    sentinel.print();
}
sentinel.reversePrint();
```

Output:

sentinel.print():

2 1 0 3
2 1 0 3 4
2 1 0 3 4 5

sentinel.reversePrint():

5 4 3 0 1 2

```

for(int i=0;i<7;i++){
    if(i%2==0){
        sentinel.pop_front();
    }
    else{
        sentinel.pop_back();
    }
    sentinel.print();
}
sentinel.reversePrint();

```

Output:

sentinel.print():

1 0 3 4 5

1 0 3 4

0 3 4

0 3

3

nothing

empty list

sentinel.reversePrint():

empty list

```

for(int i=0;i<3;i++){
    sentinel.push_back(i);
    sentinel.print();
}
sentinel.reversePrint();

```

Output:

sentinel.print():

0

0 1

0 1 2

sentinel.reversePrint():

2 1 0

```

for(int i=3;i<6;i++){
    sentinel.push_front(i);
    sentinel.print();
}
sentinel.reversePrint();

```

Output:

sentinel.print():

3 0 1 2

4 3 0 1 2

5 4 3 0 1 2

sentinel.reversePrint():

2 1 0 3 4 5

```
for(int i=0;i<7;i++){  
    if(i%2){  
        sentinel.pop_front();  
    }  
    else{  
        sentinel.pop_back();  
    }  
    sentinel.print();  
}  
sentinel.reversePrint();
```

Output:

sentinel.print():

5 4 3 0 1 2

4 3 0 1 2

4 3 0 1

3 0 1

3 0

0

nothing

empty list

sentinel.reversePrint():

empty list

```
for(int i=0;i<3;i++){  
    sentinel.push_front(i);  
    sentinel.print();  
}  
sentinel.reversePrint();
```

Output:

sentinel.print():

0
1 0
2 1 0

sentinel.reversePrint():

0 1 2