Typing Machine

Contents

1 Problem Statement

Our task is to implement typing machine using doubly linked list data structure. It is just typical typing machine which can be found in many editors.

This typing machine has 3 features:

- moving cursor,
- inserting/deleting characters near cursor,
- and printing whole string.

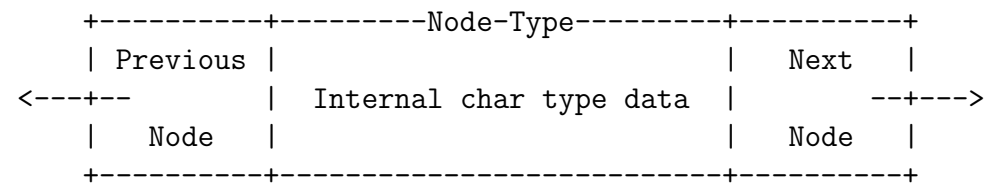
You will implement 2 classes:

- `class Node`, which is each node of linked list.
- `class TypingMachine`, which is typing machine with features.

2 Implementation

2.1 `class Node`

We will implement `Node` type denoting each node of linked list.



One `public` constructor and five `public` functions should be implemented.

- `explicit Node(char data)`: This constructor should make `Node` storing `data` as internal data.
- `char GetData()`: This function should return internal data.
- `Node* InsertPreviousNode(char data)`: This function should make new `Node` with given data and Insert between previous node and `this` node. (with `new` keyword of C++) This function should return pointer of inserted node.
- `Node* InsertNextNode(char data)`: This function should make new `Node` with given data and Insert between next node and `this` node. (with `new` keyword of C++) This function should return pointer of inserted node.
- `Node* GetPreviousNode()`: This function should return set previous node with `InsertPreviousNode` method. If not set, this function should return `nullptr`.
- `Node* GetNextNode()`: This function should return set next node with `InsertNextNode`. If not set, this function should return `nullptr`.
- `bool ErasePreviousNode()`: This function should erase previous node and deallocate. (with `delete` keyword of C++) If previous node does not exist so erase was unsuccessful, you should return `false`. Otherwise, you should return `true`.
- `bool EraseNextNode()`: This function should erase previous node and deallocate. (with `delete` keyword of C++) If next node does not exist so erase was unsuccessful, you should return `false`. Otherwise, you should return `true`.

Making another `private` functions or members are free. It is **NOT** recommended to use global variables.

You can verify correctness of this `class Node` with judging system.

2.2 `class TypingMachine`

We will implement `TypingMachine` type denoting whole typing machine.

One public constructor and seven public functions should be implemented.

- `TypingMachine()`: Initialize new Typing Machine
- `void HomeKey()`: Move cursor at the beginning of the line.
- `void EndKey()`: Move cursor at the end of the line.
- `void LeftKey()`: Move cursor one step left.
- `void RightKey()`: Move cursor one step right.
- `bool TypeKey(char key)`: Insert `key` at the left of the cursor. This typing machine only accept key in ASCII range 32 (`0x20`, ' ') through 126 (`0x7E`, '~'), inclusive. If element is out of range, you should do nothing. If you inserted a character, this function should return `true`, otherwise return `false`.
- `bool EraseKey()`: Erase a character at the left of the cursor. If it does not exist, do nothing. If you erased a character, this function should return `true`, otherwise return `false`.
- `std::string Print(char separator)`: Return typed string in typing machine. You should put `separator` to the place where cursor exists. If `separator` is `0` (`NUL` character), you should not display the cursor.

Function `LeftKey`, `RightKey`, `EraseKey` should do nothing if there does not exist target of function.

Maximum length of string in machine is 100. If there are already length 100 string in machine, `TypeKey` should do nothing and rejected.

2.3 How much LOC do I have to code?

You have to write about 100–120 lines of codes. This is statistics of model solution made with `git diff` command compared with template.

```
node.cc           | 60 ++++++++++++++++++++++++++++++++++++++++++++++++++----------
node.h            |  8 ++++++--
typing_machine.cc | 43 ++++++++++++++++++++++++++++++++++++++--------
typing_machine.h  |  7 ++++++-
4 files changed, 97 insertions(+), 21 deletions(-)
```