Dice for board games

Dice are used in many board games. From simple ones like Las Vegas and Backgammon to much more complicated like Woodcraft, Scholars of the South Tigris or Roll Player, where players must manipulate dice to win the game. Your task is to implement a set of operations allowing some manipulations of the dice.

Some players use dice trays or dice towers for dice rolling. We are going to implement a special dice tower. A player will throw some dice into the tower, but not all of the dice will pass through the tower. Some of them can be randomly kept inside and released only at some next dice throws.

You are given parts of the code of 2 classes: a Dice and a DiceTower. Extend it. Uncomment the code in the main() function to test your changes. Do not change the code in the main() function (only uncomment it). The output of your program should match the given output. Good luck!

Part 1 (1 point)

Implement constructor of the Dice class with parameters:

- _color (corresponds to color private member) - it is the color of the dice. The colors may be: red, yellow, orange, blue, and green. Use provided enum.
- _max_number (corresponds to **const** max_number private member) - it is the maximum number of the dice or in other words the number of the sides of the dice. Each dice has numbers starting from 1 and incremanted by 1, for example K6 dice has values: 1, 2, 3, 4, 5, 6. For this task assume that this value is between 4 and 12 inclusive. Do not check for correctness. **Implement the member as a const.** Hint: use (a ? b : c) operator in initialisation list. This parameter can be ommited or set to 0, in such cases set it to the value of the last created dice incremented by 1 (4 for the first created dice, and 12 if the last was 12).
- _current_number (corresponds to current_number private member) - it must be between 1 and max_number. If it exeeds these limits set it to the closest limit. The current_number may be omitted or set to 0, in such cases select the value of the dice randomly using the given randomNumberFrom1ToX() function.

Part 2 (1 point)

Implement << operator to print Dice. Print data to match given output.

Part 3 (1 point)

Implement * operator. It should reroll the dice using randomNumberFrom1ToX() function and set the new value of the dice.

Part 4 (1 point)

Implement - operator. It should combine 2 dice into another one with different color. In a game, players change 2 dice to the new one with different color. The value of the result dice is the biggest value of the combined dice. The number of sides of the new dice is equal to the number of the sides of the bigger dice (with more sides). The color of the dice is:

- red, if the first dice is orange and the second is yellow,

- yellow, if the first dice is orange and the second is red,
- yellow, if the first dice is green, and the second is blue,
- blue, if the first dice is green, and the second is yellow,
- for all other combinations take the color of the second dice.

Part 5 (1 point)

Implement ++ oparator. It increases the value of the dice by 1. If the value exceeds the number of sides, then set it to 1.

Part 6 (2 points)

Implement methods insert() and result() of the DiceTower. The first one throws a new dice into the dice tower. The dice rolls inside (call * operator of the dice). The tower always keeps 2 dice inside. If it is empty or has 1 dice inside, then the thrown dice is kept inside. If it already has 2 dice inside, then the dice colide (including the just inserted third dice), and randomly 1 is selected to fall from the tower. If the new dice stays in the tower it replaces the old dice. It is possible to throw many dice, calling multiple times insert() method. The result() method returns the sum of the dice that passed through the dice tower. **You have to use pointers in your implementation. Keep pointers to jammed dice and do not only remember their value.**

Part 7 (1 point)

Implement methods clear(), reset() of the DiceTower. The clear() method removes the dice that passed through the tower, so they are not counted by result() anymore. The reset() method removes all of the dice, also the one jammed inside of the dice tower.

Expected output

```
Dice:
red K6 with 3 pips
yellow K8 with 8 pips
green K4 with 4 pips
blue K4 with 1 pip
blue K5 with 2 pips

Rerolls:
red K6 with 2 pips
red K6 with 3 pips
yellow K8 with 2 pips
green K4 with 1 pip

Mixing colors:
blue K8 with 2 pips
yellow K4 with 1 pip
green K4 with 1 pip
red K6 with 3 pips

Incrementing dice:
red K6 with 3 pips incremented: red K6 with 4 pips
green K4 with 1 pip incremented: green K4 with 2 pips
yellow K8 with 2 pips incremented: yellow K8 with 3 pips
red K6 with 4 pips incremented twice: red K6 with 6 pips
```

```
Dice tower
inserting: red K6 with 6 pips
it rerolls to: red K6 with 3 pips
no dice in tower, dice is jammed in tower
First dice: red K6 with 3 pips, result: 0
inserting: yellow K8 with 3 pips
it rerolls to: yellow K8 with 3 pips
only 1 dice in tower, dice is jammed in tower
Second dice: yellow K8 with 3 pips, result: 0
inserting: green K4 with 2 pips
it rerolls to: green K4 with 4 pips
2 dice in tower, randomly just inserted dice pushes the second jammed dice from the
tower
Third dice: green K4 with 4 pips, result: 3
inserting: blue K4 with 1 pip
it rerolls to: blue K4 with 4 pips
2 dice in tower, randomly just inserted dice pushes the first jammed dice from the
tower
Fourth dice: blue K4 with 4 pips, result: 6
inserting: red K6 with 3 pips
it rerolls to: red K6 with 4 pips
no dice in tower, dice is jammed in tower
First dice: red K6 with 4 pips, result: 0
inserting: yellow K8 with 3 pips
it rerolls to: yellow K8 with 5 pips
only 1 dice in tower, dice is jammed in tower
Second dice: yellow K8 with 5 pips, result: 0
inserting: green K4 with 4 pips
it rerolls to: green K4 with 2 pips
2 dice in tower, randomly just inserted dice falls from the tower
Third dice: green K4 with 2 pips, result: 2
inserting: blue K4 with 4 pips
it rerolls to: blue K4 with 2 pips
2 dice in tower, randomly just inserted dice pushes the second jammed dice from the
tower
Fourth dice: blue K4 with 2 pips, result: 5
```