Source Code:

#include <iostream>

#include <string>

#include <queue>

using namespace std;

template<class T>

class Stack {

public:

Stack(){ // create an empty stack

}

bool empty(){ // returns true if the stack is empty

return (items.empty() && aux.empty());

}

void push(T item);// adds item to the top of stack

void pop();// removes an item from the stack

T top();// returns at the top of the stack

private:

queue<T> items; // holds the items you push (in reverse)

queue<T> aux; // helpful for pop and top

};

//empty stack

template<class T> void Stack<T>::push(T item) {

items.push(item);

}

template<class T> void Stack<T>::pop() {

while (size(items) != 1) {

T newValue = items.front();

items.pop();

aux.push(newValue);

}

items.pop();

while (!aux.empty()) {

items.push(aux.front());

aux.pop();

}

}

template<class T> int size(queue<T> a) {

int size = 0;

while (!a.empty()) {

size = size + 1;

a.pop();

}

return size;

}

template<class T> T Stack<T>::top() {

while (size(items) != 1) {

T newValue = items.front();

items.pop();

aux.push(newValue);

}

T value = items.front();

T newValue = items.front();

items.pop();

aux.push(newValue);

while (!aux.empty()) {

items.push(aux.front());

aux.pop();

}

return value;

}

int main() {

Stack<int> s;

for (int i = 1; i <= 10; i++) {

s.push(i);

}

while (!s.empty()) {

cout << s.top() << " ";

s.pop();

}

Stack<string> s1;

s1.push("cat");

s1.push("dog");

s1.push("frog");

s1.push("fish");

cout << endl;

while (!s1.empty()) {

cout << s1.top() << " ";

s1.pop();

}

cout << endl;

return 0;

}

Output:

