Yurii Huba (SFU ID: 301540732)
CMPT 225
Instructor: David Mitchell
2022-09-28

Testing of deque.h

Intro (methods of testing)

As class Deque is capable of creating and operating queues filled with different types of elements, I divide my testing into 3 stages. At the first stage I am performing tests of Deque filled with integers, at second stage Deque is filled with characters (letters) and at third stage Deque is filled with floating-point numbers.

First stage (integer queue) tests operating large number of elements

First step of these stage is creating a Deque and filling it with 1000 integers and viewing the queue. The output of 1000 sequential numbers proves that queue is correctly organizing large datasets of objects.

Second step is checking if Deque class can clear the queue (delete all the elements) and determine if some queue is ‘empty’ (has no elements in it). At first, I determine if full queue is recognized as not ‘empty’ (has at least one element) by condition with different outputs. Next, I clear the queue, display its content and check if it is recognized as ‘empty’. Correct outputs prove that Deque class correctly determines emptiness of any queue.

Third step is to check whether Deque can correctly add elements to existing queue. I use loop to simultaneously add 300 hundred integers in front and at the end of previously cleared queue. Displaying the contents of the queue I get numbers from 300 to 1 and immediately after them from 1 to 300. It proves that adding elements at both sides of the queue is performed correctly when the number of added elements is large.

Fourth step is determining if Deque class can remove elements from the beginning and end of the queue. To check this functions I use loop that deletes 100 elements from the both sides of the existing queue. Displaying the content of the changed queue I get numbers from 200 to 1 and from 1 to 200, which proves that functions responsible for deleting elements in Deque class correctly remove large number of elements.

Second stage (character queue) tests operating small number of elements

First step of second stage is to create a character Deque, fill it with 26 different letters and display contents of the queue, which tests if Deque can organize and view small dataset of character elements.

Second step is checking whether Deque class can distinguish small queue (small number of elements) from ‘empty’ queue and clear small queues. To check it I perform three operations: check if previously created queue is empty, clear this queue and check if is empty after clearing.

Third step is to check functions that add elements with small number of added objects. To do it I add one character in the beginning and one more at the end. After that I use a loop that adds 10 characters both in the beginning of the queue and at the end.

Fourth step is checking removal of small number of elements. Firstly I remove several elements in the loop, and after that I separately delete one character in the beginning and at the end of the queue.

Third stage (floating-point numbers)

First step is creating an array filled with 100 floating point numbers. It checks if Deque class correctly operates and views this type of data.

These stage has 3 more steps, that have the same idea as checking steps for integer queue. The steps also test large number of added and deleted elements, ‘empty’ but with floating point numbers.

The main purpose of this stage it to use large number more complicated type of elements in the queue and check all functions with it.