//COP3530 ~ Project1b ~ 26/February/2015 ~ Ricardo Stefano Reyna

//-------------------------------

//COP3530sp15_Proj1b Programmer's Guide and Report -------

//-------------------------------

//-------

// Purpose of Program

//-------

COP3503sp15_Proj1b (henceforth known as "tDeque"), is a double ended queue where you can push and pop elements into the queue.

//-------

// Command Line Options

//-------

No special options have been implemented.

//-------

// Organization of Code

//-------

The Organization of the code is as follows:

The whole code is inside the tDeque.h file since it contains the template definitions inside. Inside the header file it contains the class with it's declarations and the methods under the class declaration.

//-------

// Functions, Methods, Procedures

//-------

Inside the Deque there is an array queue, which grow and shrinks depending on the number of elements. The way elements can be inserted is by the push_front and push_back functions where they add elements to the front and back respectively. You can remove elements by the pop_front and pop_back functions. A size function is used where it returns the amount of elements inside the array. The empty function uses the size function to check if it's empty and later implemented in both pop functions.

//-------

// Efficiency

//-------

tDeque could have been more efficient if wrap around was used and given it constant time. Since it has to shift the whole array down it has a liner time.

//-------

// Known Bugs

//-------

Ocasionally it will not print the string of the pop, but it still does it though. Some times they print with no error.

//-------

// Testing

//-------

tDeque has undergone extensive testing, and the output of the program is satisfactory. As mentioned above, error handling

was of peak concern, and the program handles error and regular user input with speed and grace.

Below is an extensive testing session in which all commands were tested, as well as a plethora of errors:

lin114-09:7% make

g++ -c tDeque_main.cpp

g++ tDeque_main.o -o tDeque

lin114-09:8% ./tDeque

2

0

7

0

7

0

7

1

8

4

7

7

7

8

```
9
lin114-09:9% ./tDeque
1
0
3.2
0
1.45
4
1.45
3.2
9
lin114-09:10% ./tDeque
0
2
Caught Exception for empty stack!
0
World
1
Hello
4
World
Hello
8
lin114-09:14% ./tDeque
0
```

1

This

1

is

1

a

1

case

1

where

1

I

1

check

1

the

1

size

1

of

1

the

1

queue

4

num_emelents = 12

size_of_queue = 16

This

is

a

case

where

I

check

the

size

of

the

queue

1

I

1

grows

1

twice

1

the

1

size

4

num_emelents = 17

size_of_queue = 32

This

is

a

case

where

I

check

the

size

of

the

queue

I

grows

twice

the

size

1

and

1

shrinks

1

half

1

its

1

size

1

too

4

num_emelents = 23

size_of_queue = 32

This

is

a

case

where

I

check

the

size

of

the

queue

I

grows

twice

the

size

and

shrinks

half

its

size

too

1

this

1

is

1

just

1

filler

1

for

1

now

4

num_emelents = 29

size_of_queue = 32

This

is

a

case

where

I

check

the

size

of

the

queue

I

grows

twice

the

size

and

shrinks

half

its

size

too

this

is

just

filler

for

now

1

two

1

more

1

elements

0

BOOM

4

num_emelents = 33

size_of_queue = 64

BOOM

This

is

a

case

where

I

check

the

size

of

the

queue

I

grows

twice

the

size

and

shrinks

half

its

size

too

this

is

just

filler

for

now

two

more

elements

2

BOOM

2

This

2

is

2

a

2

case

2

where

2

I

2

check

2

the

2

size

2

of

2

the

2

queue

2

I

2

grows

2

twice

2

the

2

size

4

num_emelents = 15

size_of_queue = 64

and

shrinks

half

its

size

too

this

is

just

filler

for

now

two

more

elements

2

and

2

shrinks

2

half

2

its

2

size

2

too

2

this

4

num_emelents = 8

size_of_queue = 32

is

just

filler

for

now

two

more

elements

3

3

3

more

3

two

3

now

4

num_emelents = 3

size_of_queue = 16

is

just

filler

3

4

num_emelents = 2

size_of_queue = 8

is

just