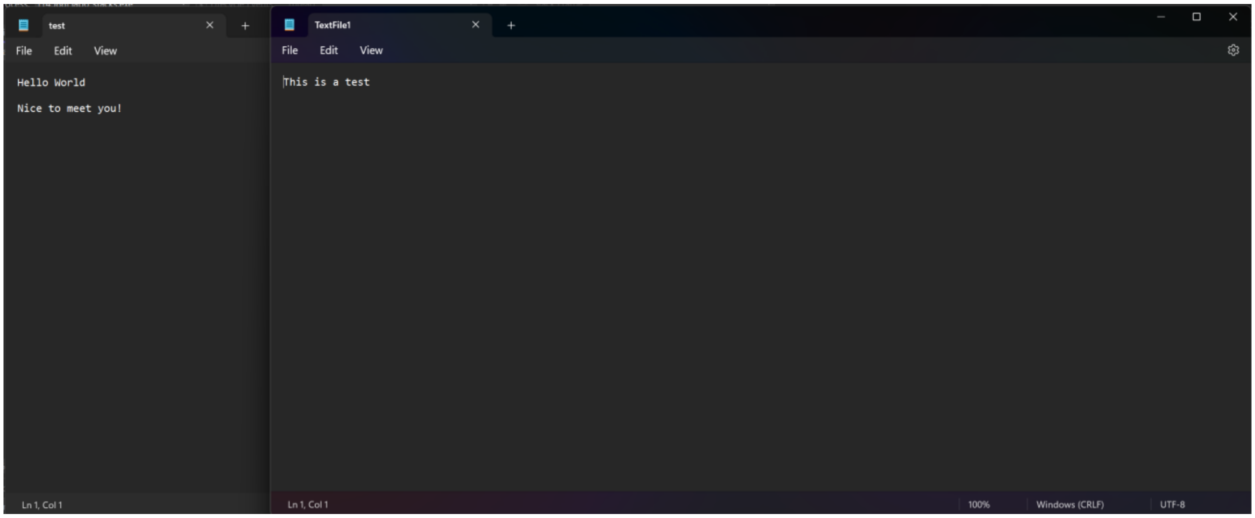
Lab 6 Report

1.

Objectives and concepts explored in this lab include classes, templates, stacks, queues and basic file I/O. It basically covered a bit of everything we have covered in class including OOP and the stacks and queues data structures to reverse a string either inputted by the user or read from a text file provided by the user. We also created custom exception classes to handle error such as overflow and underflow in stacks and queues.

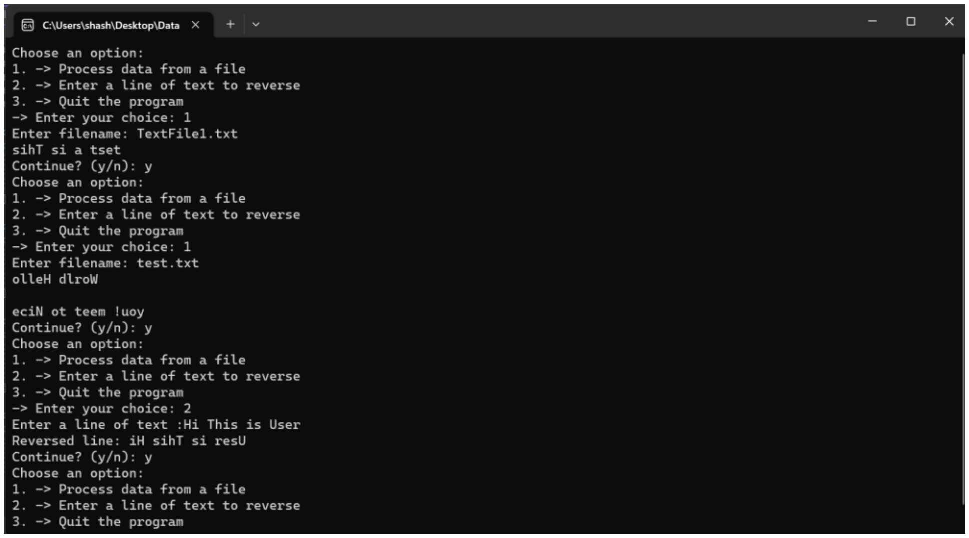
These concepts are important in software engineering as it makes sure that the program can handle errors which essential to maintain software on a large scale in the industry. Stacks and queues are widely used for software development to manage memory efficiently and has various functionalities such as when we create a To-do list online, a queue can be used to store tasks. Moreover, when have various tabs open on out browser, the history of tabs can be stored as a queue too.

Screen Shots of the output from the code:

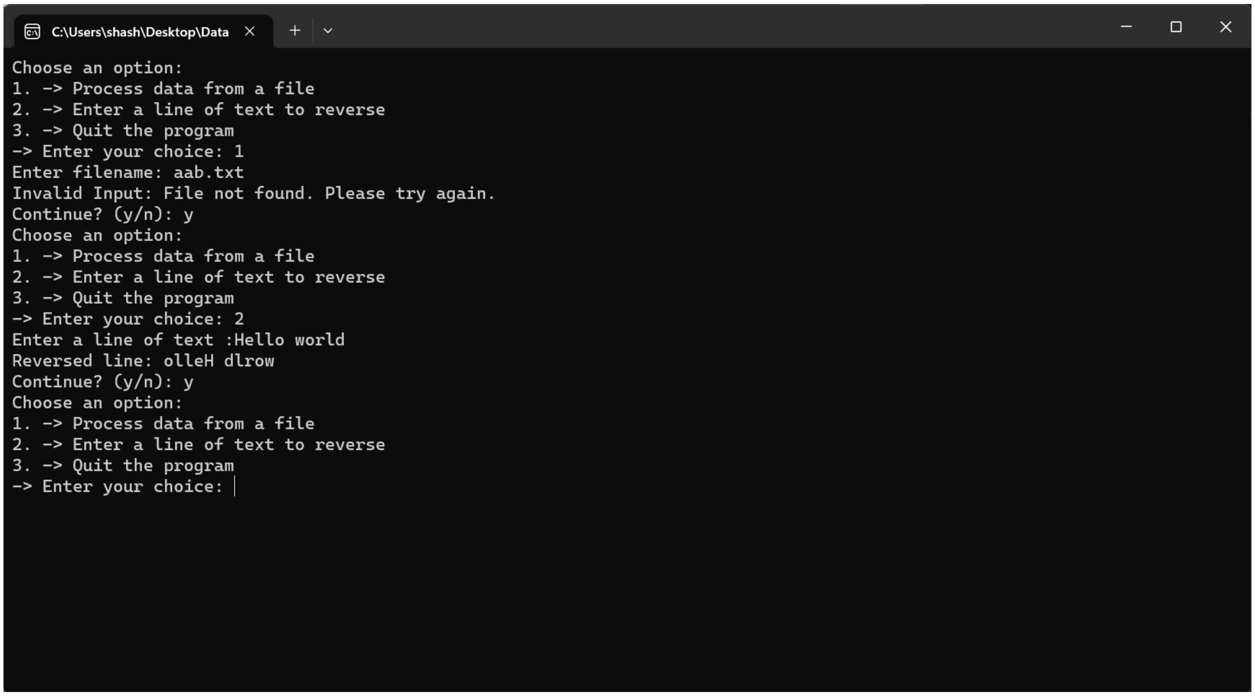


```
Hello World

Nice to meet you!
```

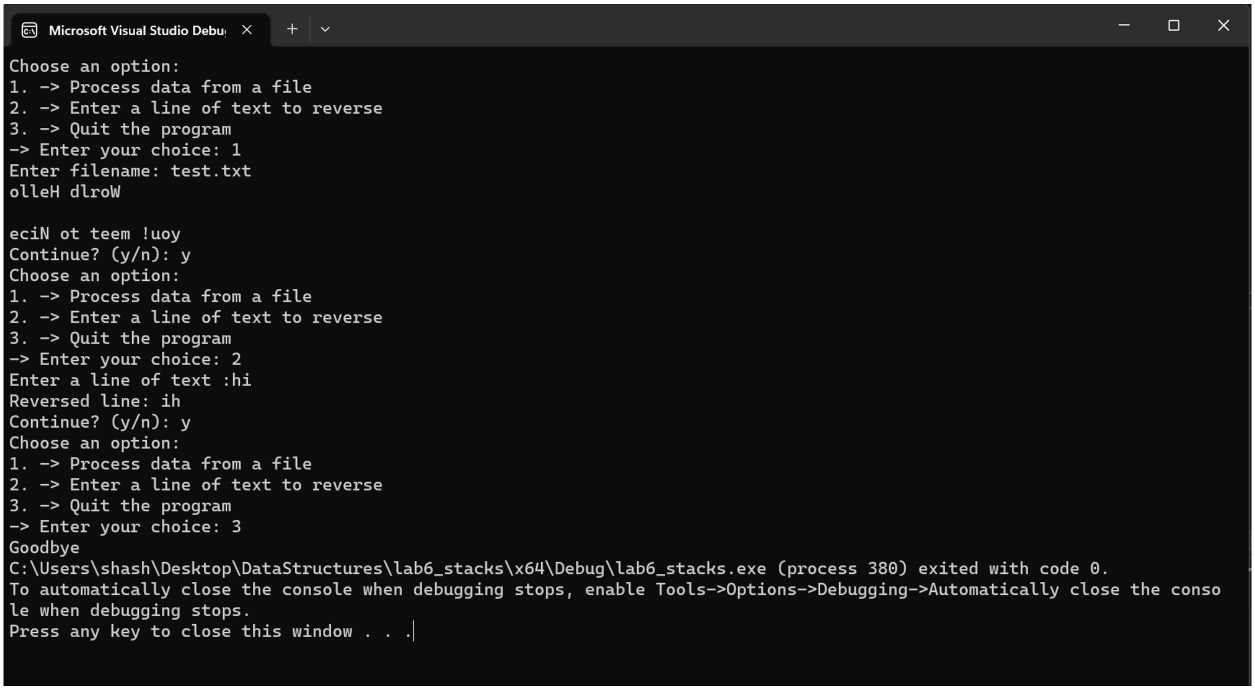
```
This is a test
```



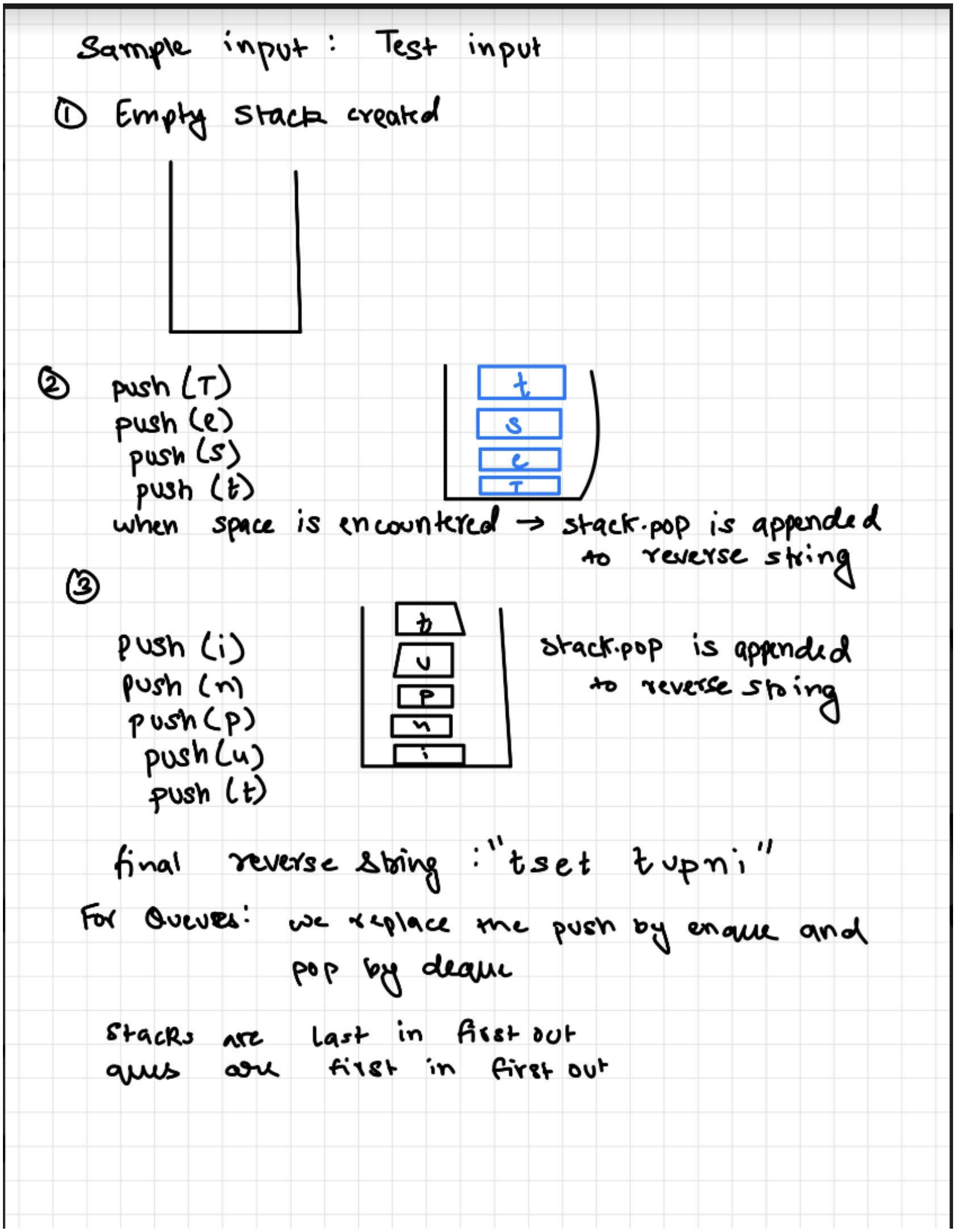
```
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 1
Enter filename: TextFile1.txt
sihT si a tset
Continue? (y/n): y
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 1
Enter filename: test.txt
olleH dlroW

eciN ot teem !uoy
Continue? (y/n): y
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 2
Enter a line of text :Hi This is User
Reversed line: iH sihT si resU
Continue? (y/n): y
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
```



```
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 1
Enter filename: aab.txt
Invalid Input: File not found. Please try again.
Continue? (y/n): y
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 2
Enter a line of text :Hello world
Reversed line: olleH dlrow
Continue? (y/n): y
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 
```



```
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 1
Enter filename: test.txt
olleH dlroW

eciN ot teem !uoy
Continue? (y/n): y
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 2
Enter a line of text :hi
Reversed line: ih
Continue? (y/n): y
Choose an option:
1. -> Process data from a file
2. -> Enter a line of text to reverse
3. -> Quit the program
-> Enter your choice: 3
Goodbye
C:\Users\shash\Desktop\DataStructures\lab6_stacks\x64\Debug\lab6_stacks.exe (process 380) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```



Sample input : Test input

① Empty stack created

② push (T)
push (e)
push (s)
push (t)

when space is encountered → stack.pop is appended to reverse string

③

push (i)
push (n)
push(p)
push(u)
push (t)

stack.pop is appended to reverse string

final reverse string :"tset tupni"

For Queues: we replace the push by enque and pop by deque

Stacks are last in first out
ques are first in first out

Diagram of stack and queue implementation

To compile this program make sure you have a text file with sample input in it for the program to read when you enter the name of the text file if you choose option 1. Make sure that text file is in the same directory of the source code. Make sure all other program files submitted are also in the same directory.