KATHMANDU UNIVERSITY

DHULIKHEL, NEPAL

COMP 202: Lab 3 Report

SUBMITTED BY

Name: Bishal Neupane

Group: Computer Engineering

Roll No: 31

Year/Semester: II/I

SUBMITTED TO

Rajani Chulyadyo, PhD

Assistant Professor

Department of Computer Science and Engineering

Date of submission: 24th September 2022

Output

Image 1: Compilation Process

Here, compilation is done through by using a shell script. The scripts accepts three arguments, one is the parent directory containing include and src folder which should be followed by -f flag, another is the build directory which should be followed by -b flag and the last one is -r flag which determines if the executable created must be executed or not.

Example usage: `./compile.sh -f lab3 -b build -r`

This will set the parent directory to `./lab3` build folder to `./build` and will run the executable after compilation (it will run because of the `-r` *flag).*

The program also utilized command line arguments passed to the main function. Here the possible arguments are:

- --type or -t which is used to specify the type of queue, if linkedlist is passed then it will use linkedlist queue else it will always use array queue.
- --size or -t which is used to specify the size of the created queue, if nothing is passed then fixed sized queue of predefined size is used in case of array queue and dynamic sized array is used which don't have fixed size in case of linked-list queue

Here, setting of the queue sized is done dynamically at runtime with Dynamic Memory Allocation.

Image 2: Solution to linking error due to templates.

The datatype of the queue is also made generic by the use of templates. And for the errors of compilation due to templates being defined in separate file is solved by including the source files in the main file where the classes are used.

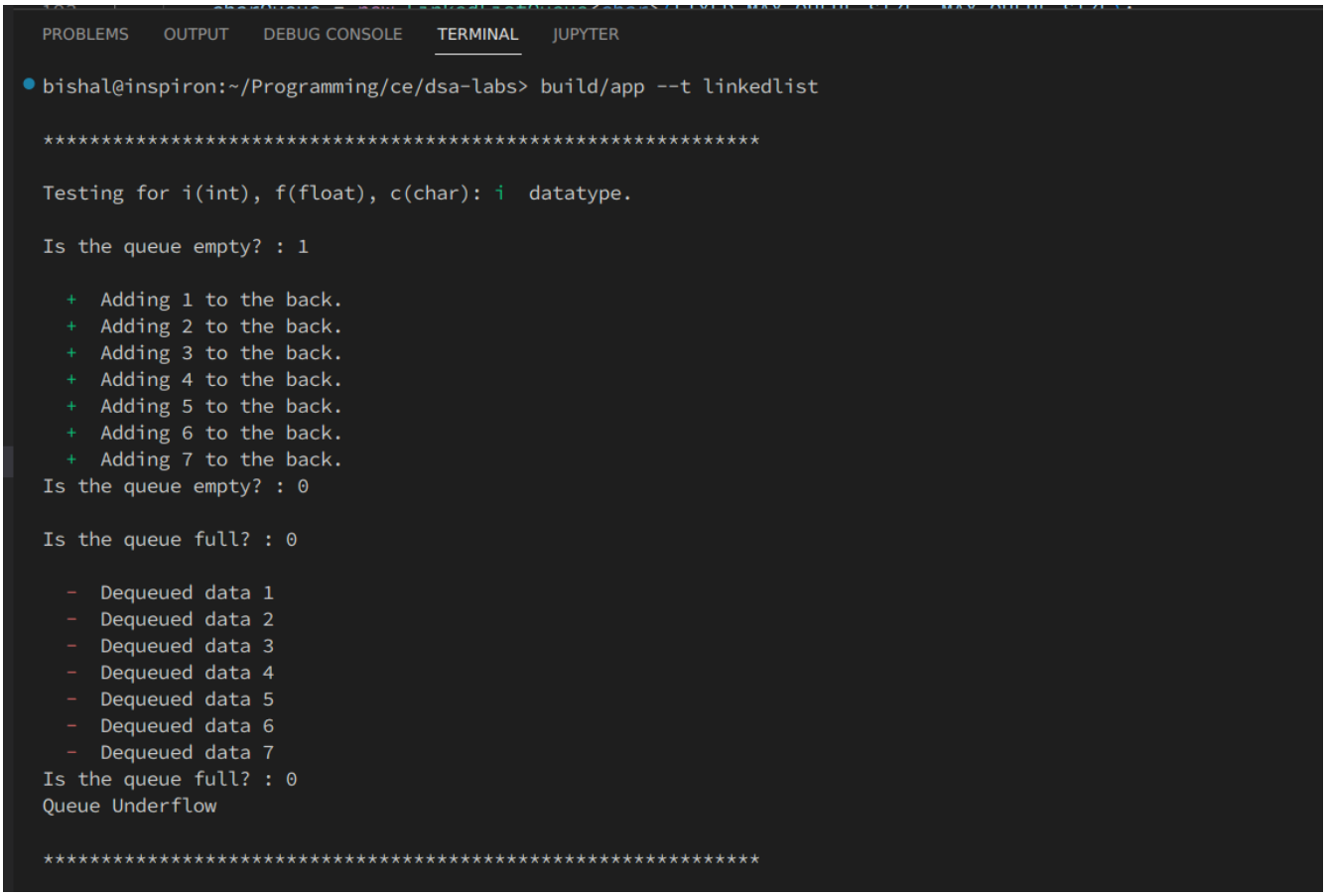

Image 3: Linked list queue for integer data type.

Here, `./build/app -t linkedlist` runs the compiled binary by setting the queue type as dynamically sized linkedlist.

The main program also consists of test functions to test the different functions of a stack. The above output shows the testing of isEmpty(), isFull(), enqueue(), dequeue(), front() and back() functions for the integer data type.

The program also implements error handling for queue underflow and queue overflow errors. The errors are caught in the local function inside main.cpp file and appropriate message is displayed.

The following images show the working of linkedlist type queue for float and char datatypes.

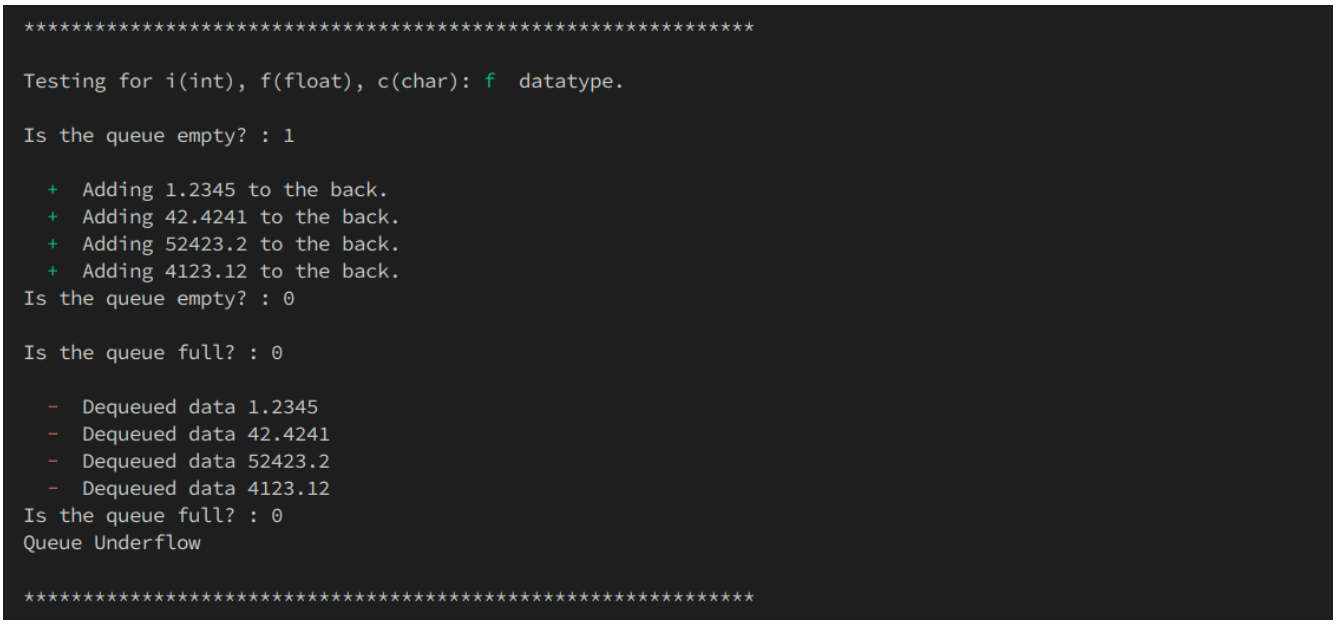

Image 4: Linked list queue for float data type.

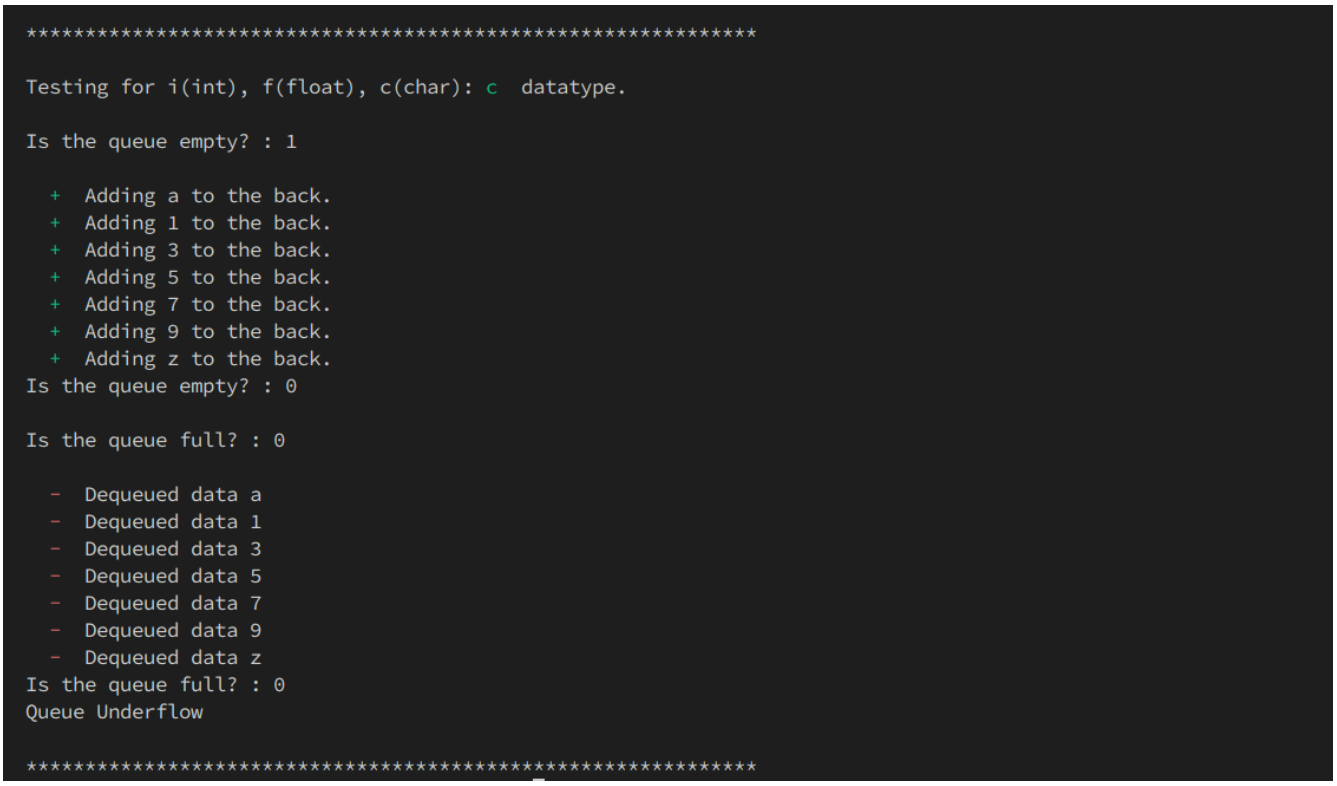

Image 5: Linked list queue for char data type.

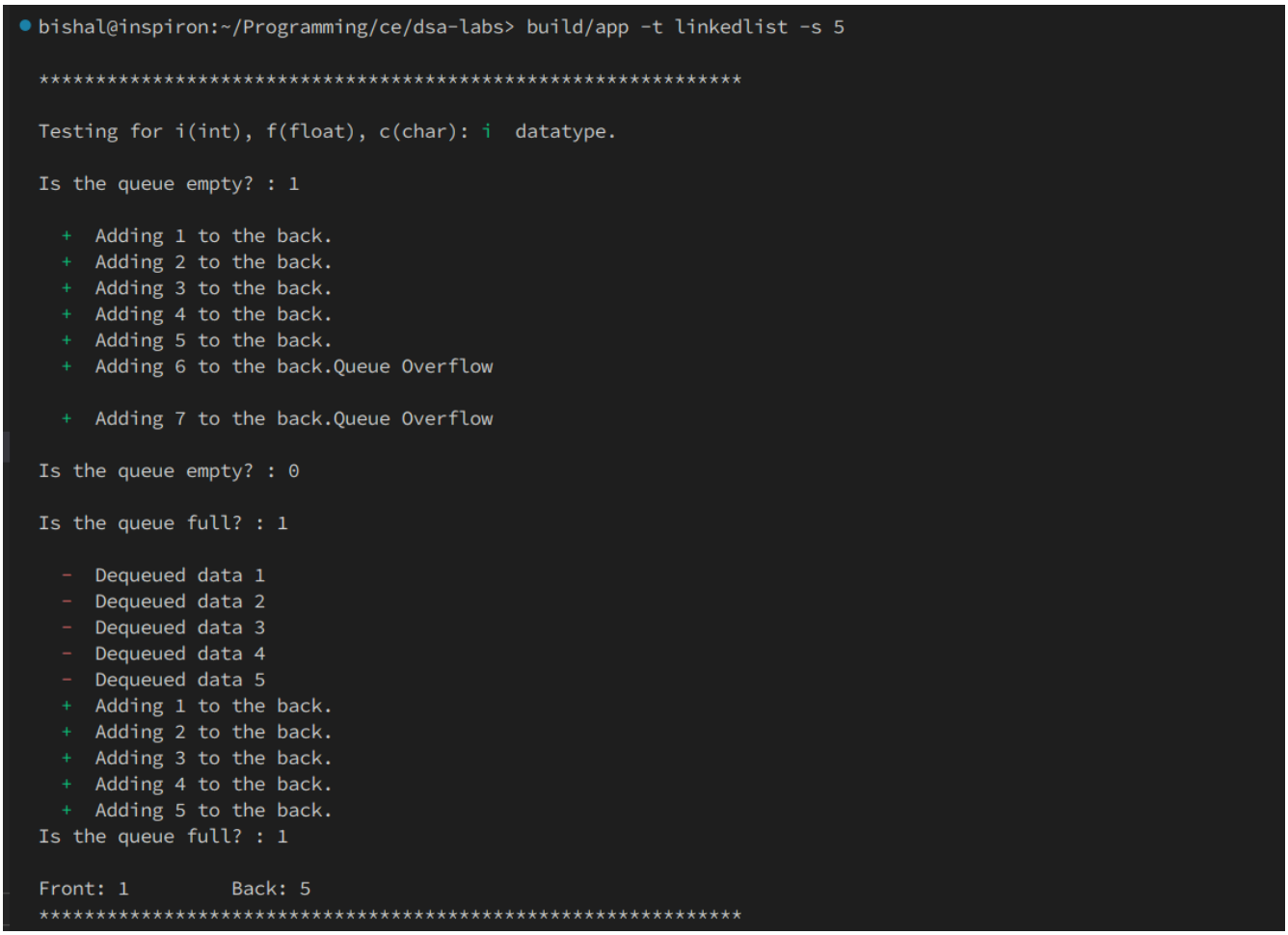

Image 6: Linked list queue of size 5

Here, `build/app -t linkedlist -s 5` runs the compiled binary by setting the stack type as fixed sized linked-list stack of size 5.

The last isFull() function returns true as the stack is fixed sized and after pushing 5 elements, it is full.

Any subsequent efforts to push element into the stack will result in queue overflow error.

```
************************************************************
bishal@inspiron:~/Programming/ce/dsa-labs> build/app -s 5

************************************************************

Testing for i(int), f(float), c(char): i  datatype.

Is the queue empty? : 1

  +  Adding 1 to the back.
  +  Adding 2 to the back.
  +  Adding 3 to the back.
  +  Adding 4 to the back.
  +  Adding 5 to the back.
  +  Adding 6 to the back.Queue Overflow

  +  Adding 7 to the back.Queue Overflow

Is the queue empty? : 0

Is the queue full? : 1

  -  Dequeued data 1
  -  Dequeued data 2
  -  Dequeued data 3
  -  Dequeued data 4
  -  Dequeued data 5
  +  Adding 1 to the back.
  +  Adding 2 to the back.
  +  Adding 3 to the back.
  +  Adding 4 to the back.
  +  Adding 5 to the back.
Is the queue full? : 1

Front: 1         Back: 5
************************************************************
```

Image 7: Array queue of size 5

Here, `./build/app --s 5` runs the compiled binary by setting the queue type as fixed size array queue of size 5. If anything is passed in `--type` or `-t` argument other than `linkedlist`, array queue is used.

The setting of size of array queue is also done at runtime through DMA, this makes it possible to change the size of queue without recompiling the program files.