Exercise 1

a will remain the same while b will be changed. The reason being that x was passed by value and b was passed by reference, meaning the function could change the original value.

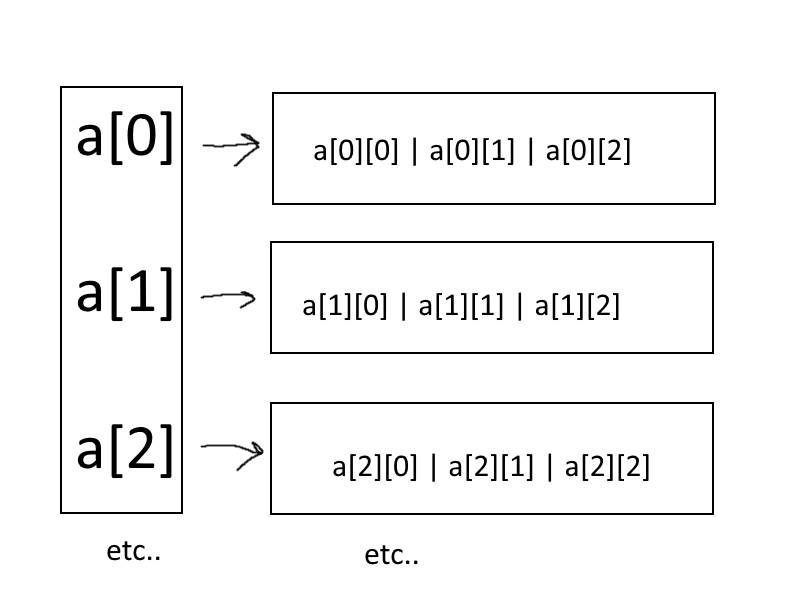
The output produced will be “2 21 3.”

Exercise 2

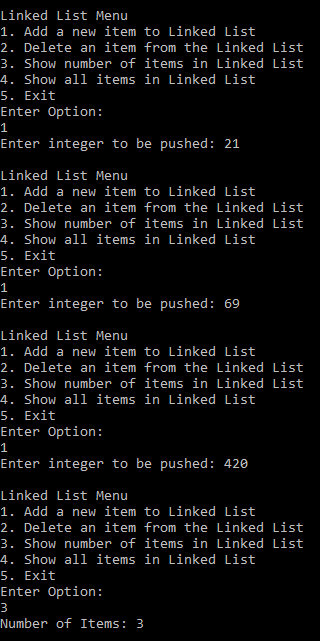
1. The compiler needs the size to be able to do pointer arithmetic correctly. Int array[2] [5] becomes int (\*array) [5].
2. Int \*\*A = new int\*[3]

for(int i = 0;i < 3; i++)

A[i] = new int[5]

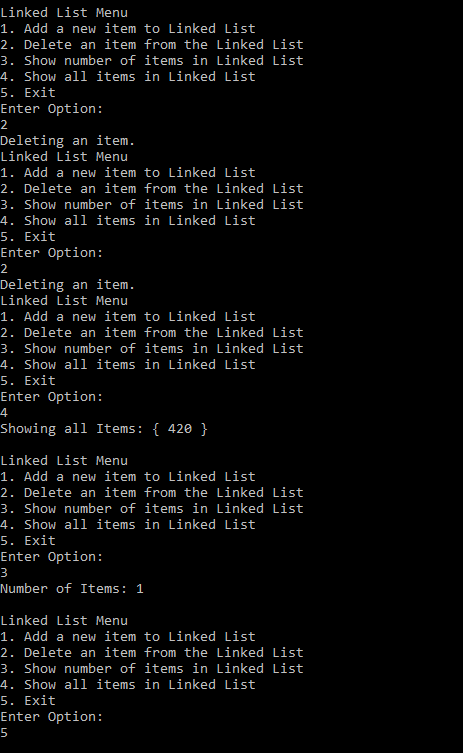


Exercise 3



I added three items to the linked list

Showing the number of items is 3



I popped/deleted an item from the linked list

Another one is popped

Showing the queue to indicate the items were indeed popped

Shows the number of items

Exits the program

g++ main.cpp LinkedList.h

Exercise 4:

**Errors are bolded**

#include <iostream>

**//include namespace std**

int main()

{

int\* ptr;

int\* temp;

int x;

ptr = new int;

\*ptr = 4;

\*temp = \*ptr;

cout << ptr << temp;

x = 9;

\*temp = x;

cout << \*ptr << \*temp;

ptr = new int;

ptr = 5; **//Must change ptr to \*ptr, can’t convert into int\***

cout << \*ptr << \*temp; **//output is : 5 9**

return 0; **// no deletion of ptr, aka delete [] ptr**

}