Christopher Bass

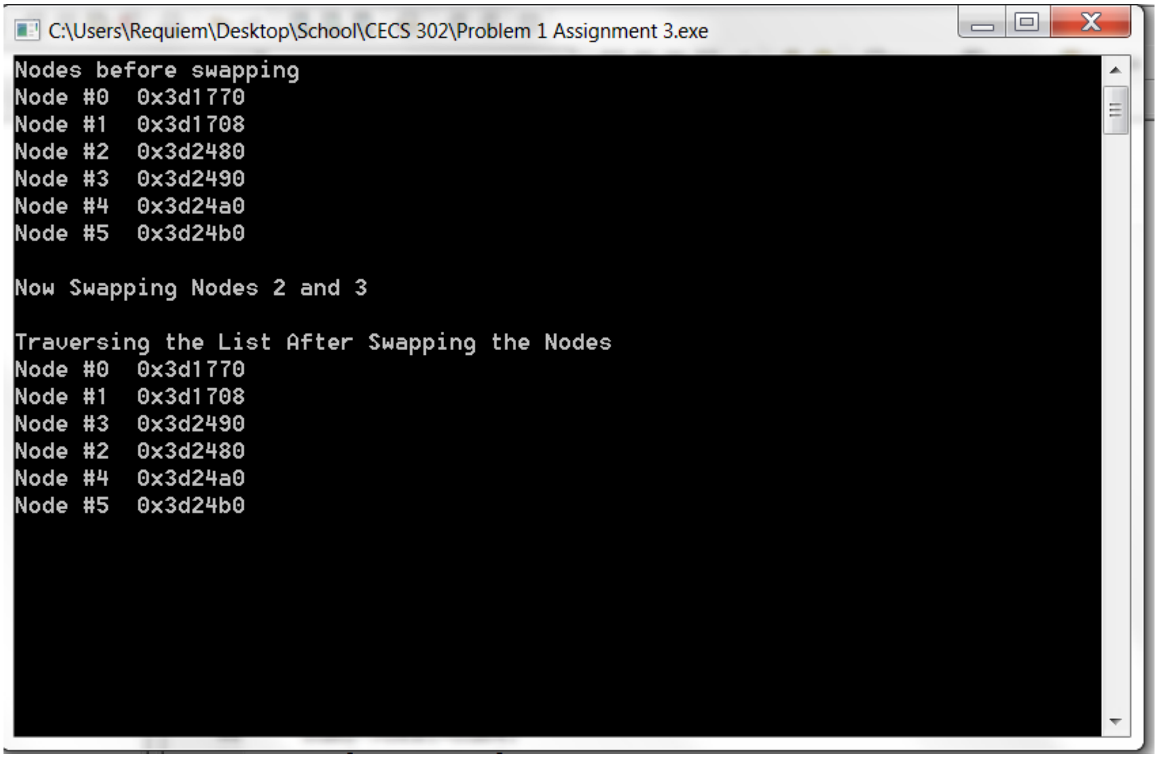
CECS 302 50

Assignment 3

10/20/14

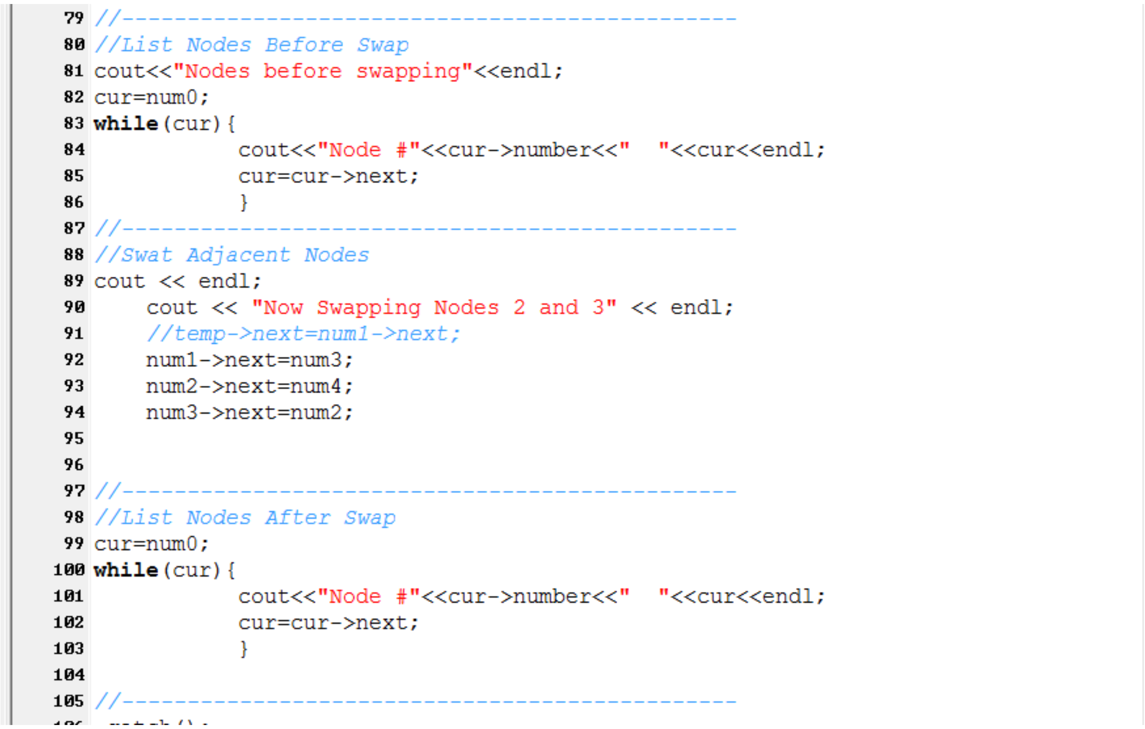
3.2 Swap two adjacent elements by adjusting only the links (and not the data) using:

a. Singly linked list



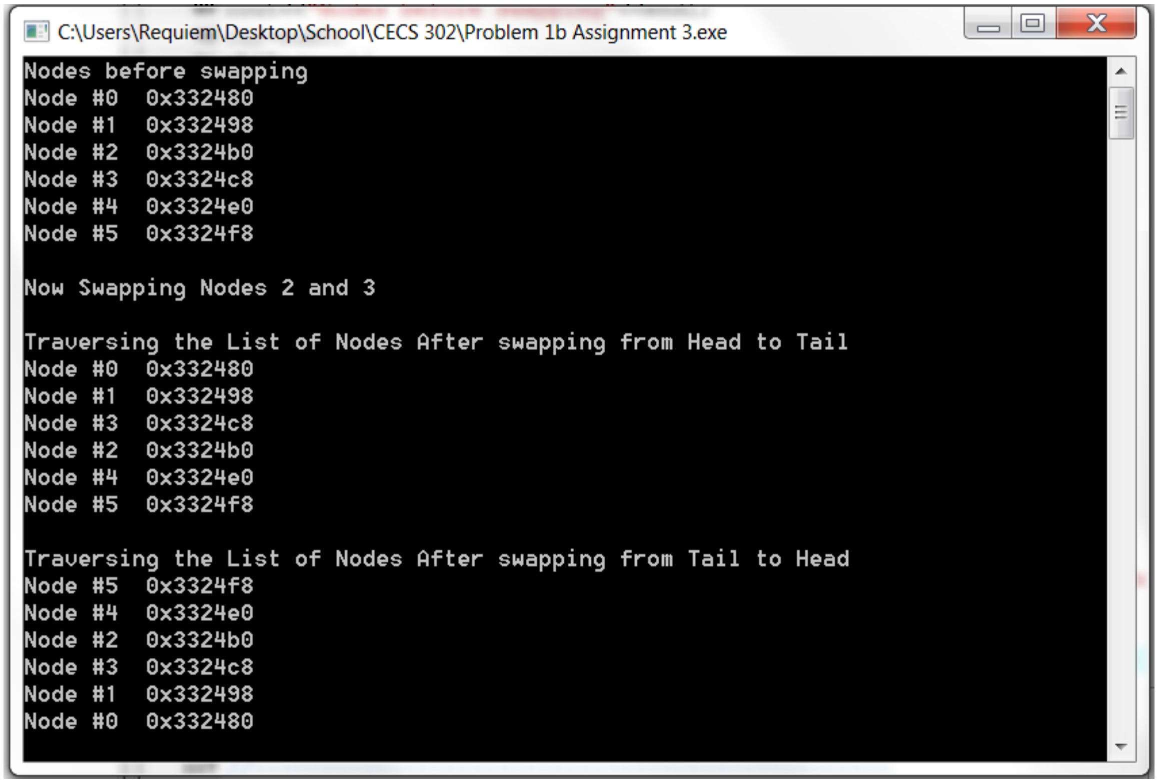
C:\Users\Requiem\Desktop\School\CECS 302\Problem 1 Assignment 3.exe

```
Nodes before swapping
Node #0  0x3d1770
Node #1  0x3d1708
Node #2  0x3d2480
Node #3  0x3d2490
Node #4  0x3d24a0
Node #5  0x3d24b0

Now Swapping Nodes 2 and 3

Traversing the List After Swapping the Nodes
Node #0  0x3d1770
Node #1  0x3d1708
Node #3  0x3d2490
Node #2  0x3d2480
Node #4  0x3d24a0
Node #5  0x3d24b0
```



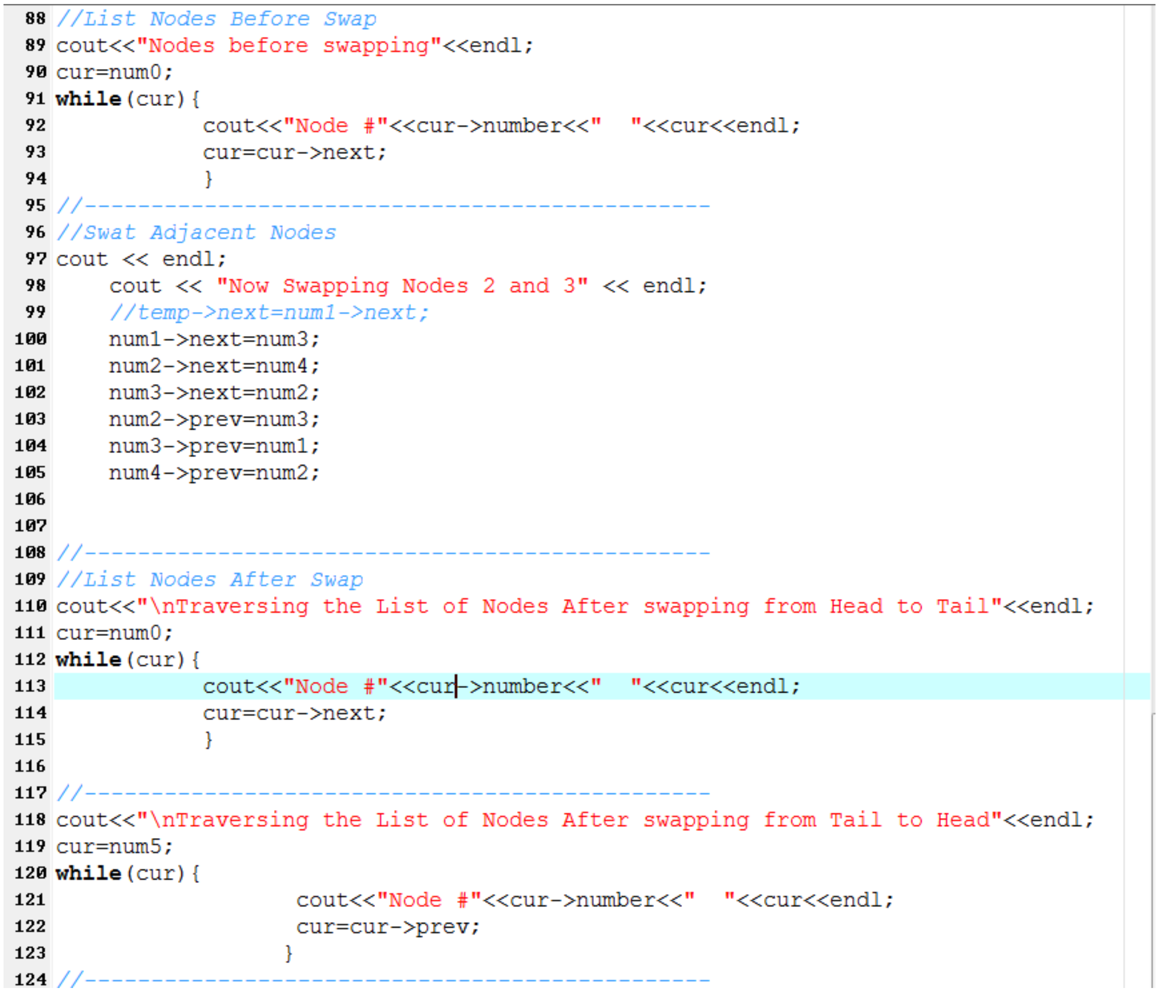
```
//-----------------------------------------------
//List Nodes Before Swap
cout<<"Nodes before swapping"<<endl;
cur=num0;
while(cur){
           cout<<"Node #"<<cur->number<<"  "<<cur<<endl;
           cur=cur->next;
           }
//-----------------------------------------------
//Swat Adjacent Nodes
cout << endl;
    cout << "Now Swapping Nodes 2 and 3" << endl;
    //temp->next=num1->next;
    num1->next=num3;
    num2->next=num4;
    num3->next=num2;


//-----------------------------------------------
//List Nodes After Swap
cur=num0;
while(cur){
           cout<<"Node #"<<cur->number<<"  "<<cur<<endl;
           cur=cur->next;
           }

//-----------------------------------------------
 getch();
```

b.) Doubly Linked List after swapping I use a loop to traverse the list from both Head to Tail and Tail to Head to show that all the pointers are working properly.



C:\Users\Requiem\Desktop\School\CECS 302\Problem 1b Assignment 3.exe

```
Nodes before swapping
Node #0  0x332480
Node #1  0x332498
Node #2  0x3324b0
Node #3  0x3324c8
Node #4  0x3324e0
Node #5  0x3324f8

Now Swapping Nodes 2 and 3

Traversing the List of Nodes After swapping from Head to Tail
Node #0  0x332480
Node #1  0x332498
Node #3  0x3324c8
Node #2  0x3324b0
Node #4  0x3324e0
Node #5  0x3324f8

Traversing the List of Nodes After swapping from Tail to Head
Node #5  0x3324f8
Node #4  0x3324e0
Node #2  0x3324b0
Node #3  0x3324c8
Node #1  0x332498
Node #0  0x332480
```



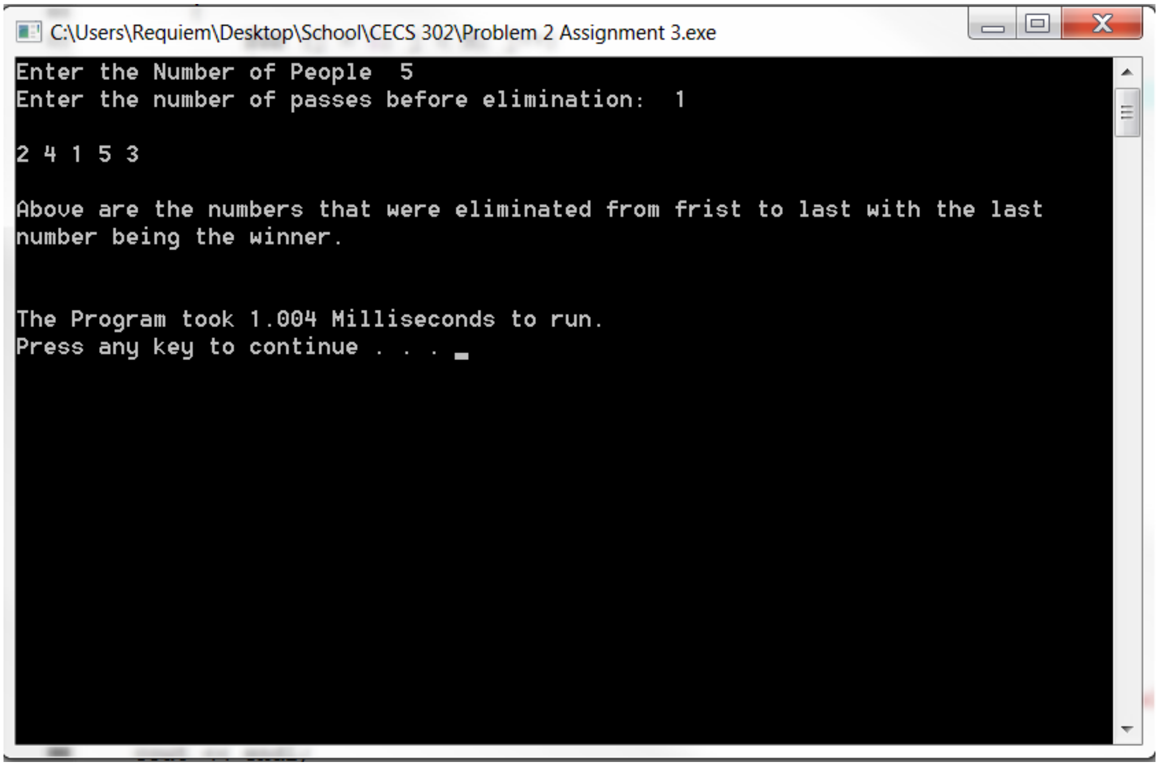
```
//List Nodes Before Swap
cout<<"Nodes before swapping"<<endl;
cur=num0;
while(cur){
           cout<<"Node #"<<cur->number<<"  "<<cur<<endl;
           cur=cur->next;
           }
//---------------------------------------------------
//Swat Adjacent Nodes
cout << endl;
    cout << "Now Swapping Nodes 2 and 3" << endl;
    //temp->next=num1->next;
    num1->next=num3;
    num2->next=num4;
    num3->next=num2;
    num2->prev=num3;
    num3->prev=num1;
    num4->prev=num2;


//---------------------------------------------------
//List Nodes After Swap
cout<<"\nTraversing the List of Nodes After swapping from Head to Tail"<<endl;
cur=num0;
while(cur){
           cout<<"Node #"<<cur->number<<"  "<<cur<<endl;
           cur=cur->next;
           }

//---------------------------------------------------
cout<<"\nTraversing the List of Nodes After swapping from Tail to Head"<<endl;
cur=num5;
while(cur){
                  cout<<"Node #"<<cur->number<<"  "<<cur<<endl;
                  cur=cur->prev;
                 }
//---------------------------------------------------
```

3.6 The Josephus problem is the following game: N people, numbered 1 to N, are sitting in a circle. Starting at person 1, a hot potato is passed. After M passes, the person holding the hot potato is eliminated, the circle closes ranks, and the game continues with the person who was sitting after the eliminated person picking up the hot potato. The last remaining person wins. Thus, if M=0 and N=5, players are eliminated in order, and player 5 wins. If M=1 and N=5, the order of the elimination is 2, 4, 1, 5.

a.) Write a program to solve the Josephus problem for general values of M and N. Try to make your program as efficient as possible. Make sure you dispose of cells.

b.) What is the running time of your program.

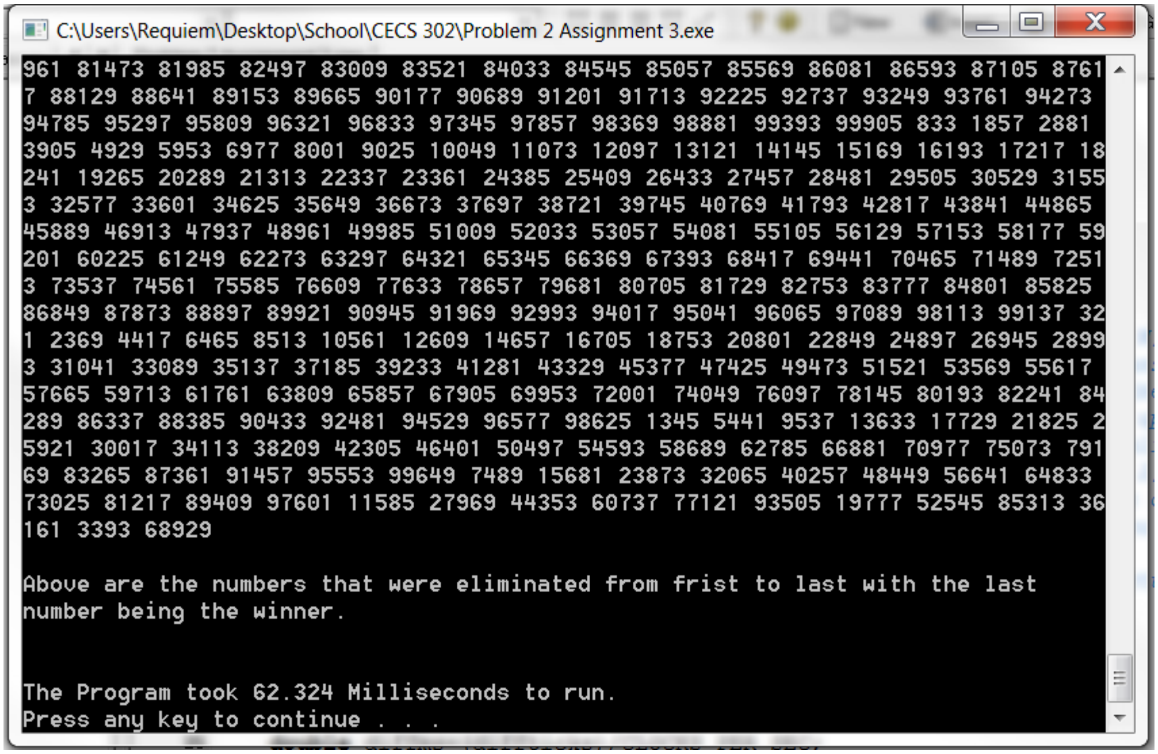
c.) If M=1, what is the running time of your program? How is the actual speed affected by the delete routine for large values of N (N > 100,000)?



C:\Users\Requiem\Desktop\School\CECS 302\Problem 2 Assignment 3.exe

```
Enter the Number of People  5
Enter the number of passes before elimination:  1

2 4 1 5 3

Above are the numbers that were eliminated from frist to last with the last
number being the winner.


The Program took 1.004 Milliseconds to run.
Press any key to continue . . .
```

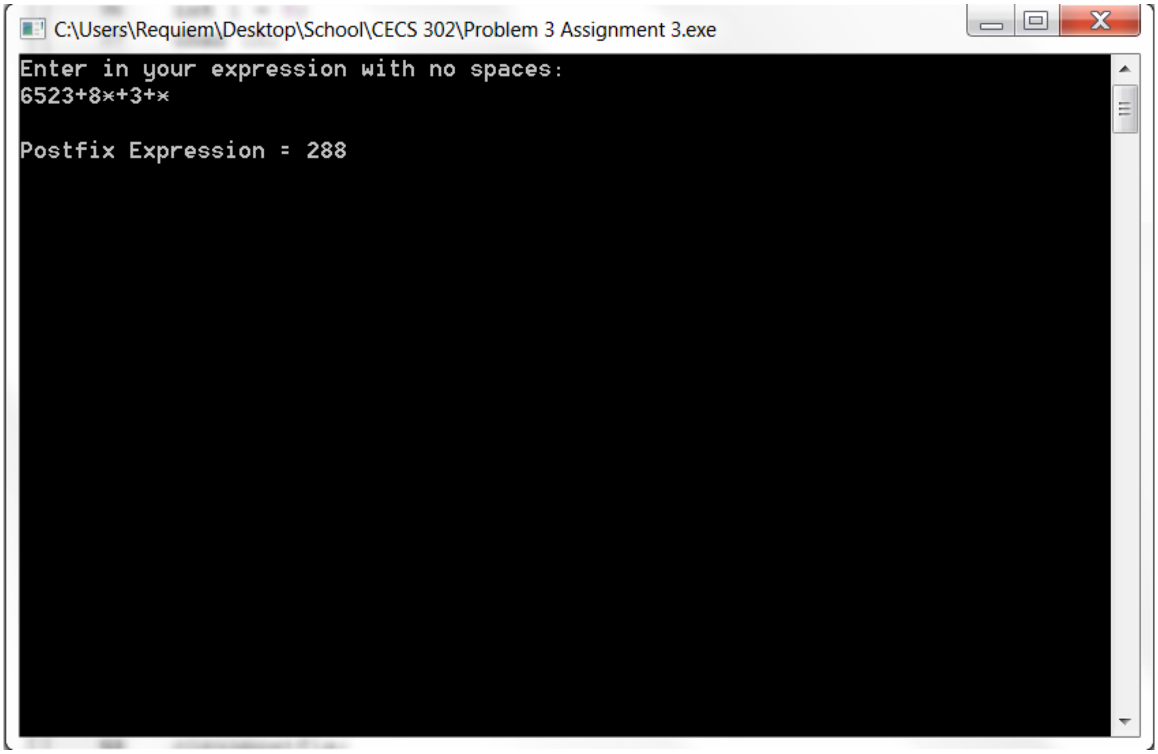
c.) M=1, N=100,000 Takes much longer due to recursion.



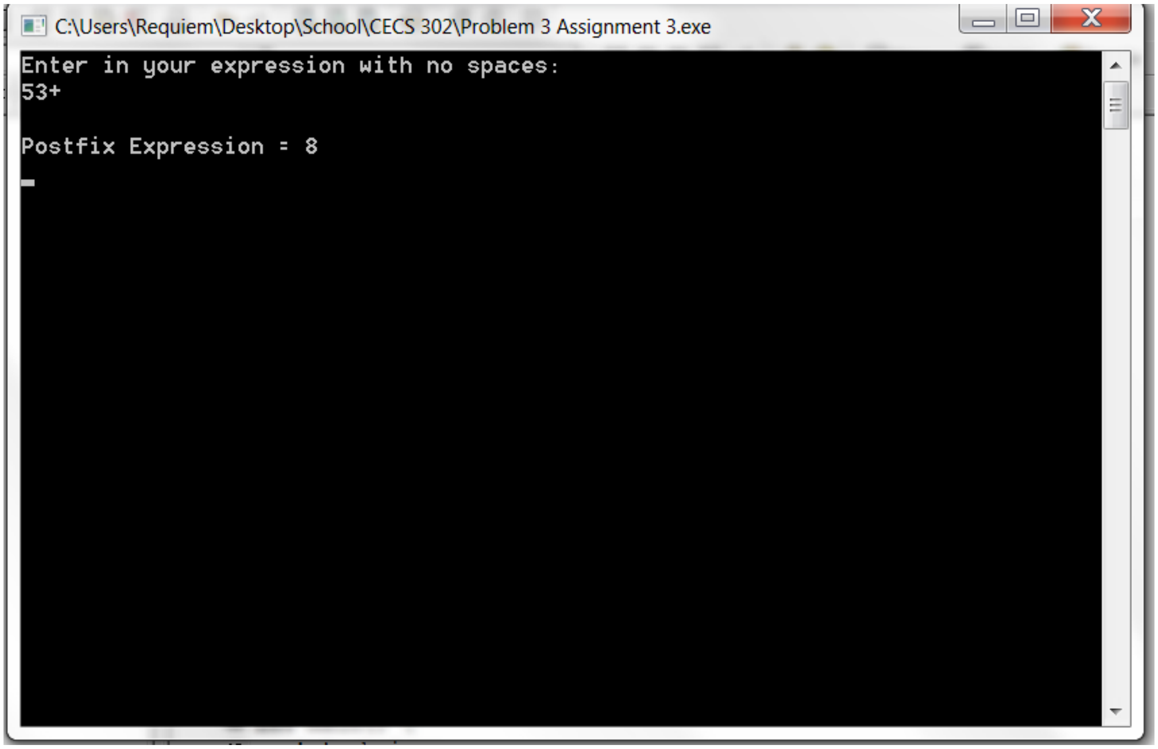
C:\Users\Requiem\Desktop\School\CECS 302\Problem 2 Assignment 3.exe

```
961 81473 81985 82497 83009 83521 84033 84545 85057 85569 86081 86593 87105 8761
7 88129 88641 89153 89665 90177 90689 91201 91713 92225 92737 93249 93761 94273
94785 95297 95809 96321 96833 97345 97857 98369 98881 99393 99905 833 1857 2881
3905 4929 5953 6977 8001 9025 10049 11073 12097 13121 14145 15169 16193 17217 18
241 19265 20289 21313 22337 23361 24385 25409 26433 27457 28481 29505 30529 3155
3 32577 33601 34625 35649 36673 37697 38721 39745 40769 41793 42817 43841 44865
45889 46913 47937 48961 49985 51009 52033 53057 54081 55105 56129 57153 58177 59
201 60225 61249 62273 63297 64321 65345 66369 67393 68417 69441 70465 71489 7251
3 73537 74561 75585 76609 77633 78657 79681 80705 81729 82753 83777 84801 85825
86849 87873 88897 89921 90945 91969 92993 94017 95041 96065 97089 98113 99137 32
1 2369 4417 6465 8513 10561 12609 14657 16705 18753 20801 22849 24897 26945 2899
3 31041 33089 35137 37185 39233 41281 43329 45377 47425 49473 51521 53569 55617
57665 59713 61761 63809 65857 67905 69953 72001 74049 76097 78145 80193 82241 84
289 86337 88385 90433 92481 94529 96577 98625 1345 5441 9537 13633 17729 21825 2
5921 30017 34113 38209 42305 46401 50497 54593 58689 62785 66881 70977 75073 791
69 83265 87361 91457 95553 99649 7489 15681 23873 32065 40257 48449 56641 64833
73025 81217 89409 97601 11585 27969 44353 60737 77121 93505 19777 52545 85313 36
161 3393 68929

Above are the numbers that were eliminated from frist to last with the last
number being the winner.


The Program took 62.324 Milliseconds to run.
Press any key to continue . . .
```

3.22 Write a program to evaluate a postfix expression


C:\Users\Requiem\Desktop\School\CECS 302\Problem 3 Assignment 3.exe

```
Enter in your expression with no spaces:
6523+8*+3+*

Postfix Expression = 288
```


C:\Users\Requiem\Desktop\School\CECS 302\Problem 3 Assignment 3.exe

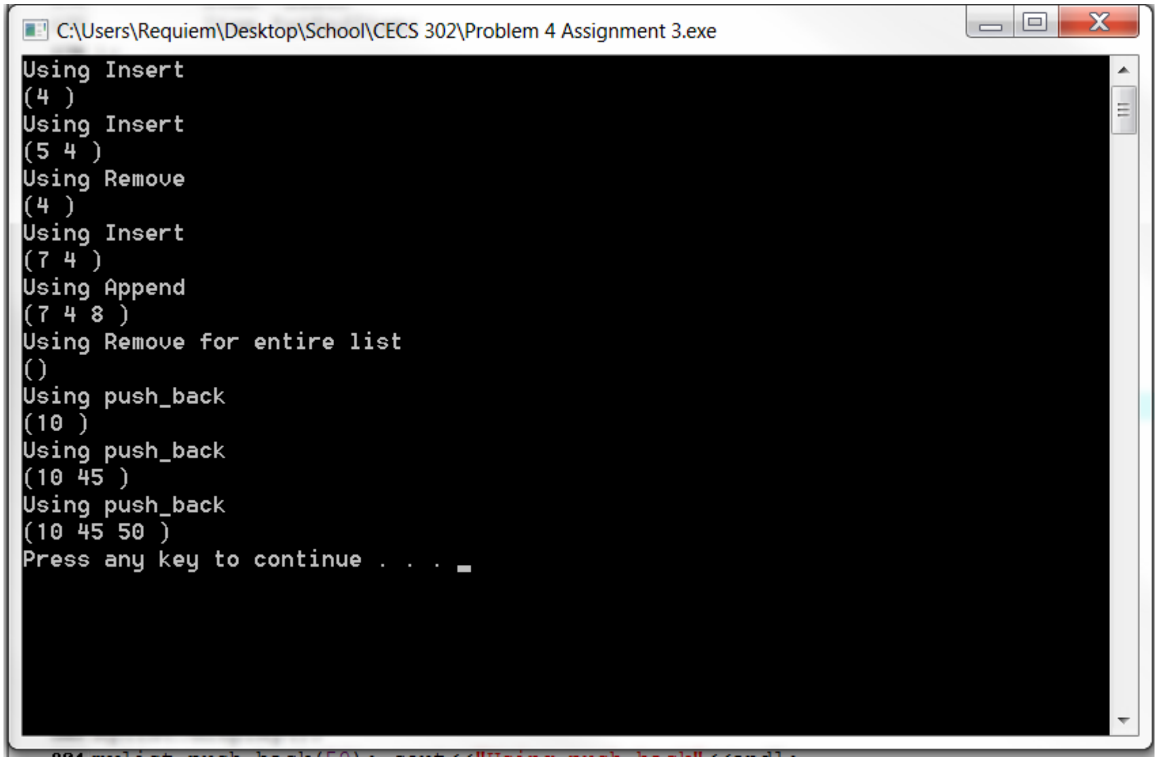
```
Enter in your expression with no spaces:
53+

Postfix Expression = 8
```

4.) Starting with my code for a single linked list defined in SLList.h and implemented in SLList.cpp, do the following:

1.) Write an additional method called push_back (type someItem) that will add   someItem to the end of the list.

2.) Write your own main that demonstrates the use of every public method.

3.) Compare append with push_back for large number of points and try to obtain some empirical estimates using the timing functions I discussed in chapter 2 in MaxSubSum example which is posted on blackboard.



C:\Users\Requiem\Desktop\School\CECS 302\Problem 4 Assignment 3.exe

```
Using Insert
(4 )
Using Insert
(5 4 )
Using Remove
(4 )
Using Insert
(7 4 )
Using Append
(7 4 8 )
Using Remove for entire list
()
Using push_back
(10 )
Using push_back
(10 45 )
Using push_back
(10 45 50 )
Press any key to continue . . .
```