

|  |    |
|--|----|
| Figure 1: initial running the program, showing the content of the file read stored in array and presenting user options..... | 2  |
| Figure 2: list menu and creation option .....  | 2  |
| Figure 3: view list.....   | 3  |
| Figure 4: dynamic adding of new element to the list, note that the original array wont change ..                             | 3  |
| Figure 5: deleting student with id 22215 and viewing the list again to confirm.....  | 4  |
| Figure 6: deleting students with GPA less than certain value, showing updated list .....                                     | 5  |
| Figure 7: double linked list created from original student array.....  | 6  |
| Figure 8: deleting a student using first and last name .....   | 7  |
| Figure 9: delete uninsured students .....  | 7  |
| Figure 10: ordered queue created, sorted by id from highest to lowest.....   | 8  |
| Figure 11: sample dequeue of some students .....   | 9  |
| Figure 12: double linked stack menu, push example .....  | 10 |
| Figure 13: push top example .....  | 10 |
| Figure 14: pop examples .....  | 11 |
| Figure 15: binary tree creation and preorder view (ordering is based on fname).....  | 12 |
| Figure 16: post order print of left balanced binary tree .....   | 13 |
| Figure 17: adjacency matrix .....  | 13 |

```

name1 lname1 22211 77 Math 1
name2 lname2 12212 78 Math 1
name3 lname3 22213 79 physics 1
name5 lname5 22215 78 physics 0
name7 lname7 22217 58 Math 1
name4 lname4 12214 77 Math 0
name6 lname6 12216 66 Math 1

```

Data.txt file content

## MAIN OPTIONS

```
***** CE 368, C Data Structures, Lists, Queues, Binary Trees, Stacks *****
===== Student File was read into Array with Content:
  name1   lname1   22211   77   Math   1
  name2   lname2   12212   78   Math   1
  name3   lname3   22213   79   physics  1
  name5   lname5   22215   78   physics  0
  name7   lname7   22217   58   Math    1
  name4   lname4   12214   77   Math    0
  name6   lname6   12216   66   Math    1
=====
*****MAIN MENU *****
0: Exit
1:Linked List Menu
2:Doubly Linked List Menu
3:Linked Queue Menu
4:Doubly Linked Stack Menu
5:Binary Tree Menu
input a choice:
```

Figure 1: initial running the program, showing the content of the file read stored in array and presenting user options

## LINKED LIST OPTIONS

```
*****MAIN MENU *****
0: Exit
1:Linked List Menu
2:Doubly Linked List Menu
3:Linked Queue Menu
4:Doubly Linked Stack Menu
5:Binary Tree Menu
input a choice:1

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:1
Linked List Created ...
```

Figure 2: list menu and creation option

```

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:2
===== Linked list =====
    name6    lname6    12216    66    Math    1
    name4    lname4    12214    77    Math    0
    name7    lname7    22217    58    Math    1
    name5    lname5    22215    78    physics  0
    name3    lname3    22213    79    physics  1
    name2    lname2    12212    78    Math    1
    name1    lname1    22211    77    Math    1
=====

```

Figure 3: view list

```

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:3
input first name:testname
input last name:testlastname
input id:22
input GPA:55
input major:it
input insurance status (1 for insured) (0 for uninsured):0

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:2
===== Linked list =====
    testname testlastname    22    55    it    0
    name6    lname6    12216    66    Math    1
    name4    lname4    12214    77    Math    0
    name7    lname7    22217    58    Math    1
    name5    lname5    22215    78    physics  0
    name3    lname3    22213    79    physics  1
    name2    lname2    12212    78    Math    1
    name1    lname1    22211    77    Math    1
=====

```

Figure 4: dynamic adding of new element to the list, note that the original array wont change

```

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:4
input student id:22215
student record with id [22215] was deleted from linked list

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:2
===== Linked list =====
  testname testlastname      22    55      it      0
    name6      lname6      12216    66      Math      1
    name4      lname4      12214    77      Math      0
    name7      lname7      22217    58      Math      1
    name3      lname3      22213    79      physics    1
    name2      lname2      12212    78      Math      1
    name1      lname1      22211    77      Math      1
=====

```

Figure 5: deleting student with id 22215 and viewing the list again to confirm

```

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:5
input GPA:70
student record with id [22] gpa [55] was deleted from linked list
student record with id [12216] gpa [66] was deleted from linked list
student record with id [22217] gpa [58] was deleted from linked list

***LIST MENU ***
0: Return To Main Menu
1: Create a Linked List of all students.
2: View the linked list
3: Insert New Student
4: Delete a student with specific ID.
5: Delete all student with GPA less than the GPA entered by the user.
input a choice:2
===== Linked list =====
    name4      lname4      12214      77      Math      0
    name3      lname3      22213      79      physics    1
    name2      lname2      12212      78      Math      1
    name1      lname1      22211      77      Math      1
=====
***LIST MENU ***

```

Figure 6: deleting students with GPA less than certain value, showing updated list

## Double LINKED LIST OPTIONS

```
***Doubly Linked LIST MENU ***
0: Return To Main Menu
1: Create a Doubly Linked List of all students.
2: View the Doubly Linked List.
3: Insert New Student
4: Delete a student whose name is specified by the user.
5: Delete all students that don't have health insurance.
input a choice:1
Double Linked List Created ...

***Doubly Linked LIST MENU ***
0: Return To Main Menu
1: Create a Doubly Linked List of all students.
2: View the Doubly Linked List.
3: Insert New Student
4: Delete a student whose name is specified by the user.
5: Delete all students that don't have health insurance.
input a choice:2
===== Double Linked list =====
    name6    lname6    12216    66    Math    1
    name4    lname4    12214    77    Math    0
    name7    lname7    22217    58    Math    1
    name5    lname5    22215    78    physics  0
    name3    lname3    22213    79    physics  1
    name2    lname2    12212    78    Math    1
    name1    lname1    22211    77    Math    1
=====

***Doubly Linked LIST MENU ***
0: Return To Main Menu
```

Figure 7: double linked list created from original student array

```

***Doubly Linked LIST MENU ***
0: Return To Main Menu
1: Create a Doubly Linked List of all students.
2: View the Doubly Linked List.
3: Insert New Student
4: Delete a student whose name is specified by the user.
5: Delete all students that don't have health insurance.
input a choice:4
input student first name:name2
input student last name:lname2
student record with name [name2 lname2] was deleted from double linked list

***Doubly Linked LIST MENU ***
0: Return To Main Menu
1: Create a Doubly Linked List of all students.
2: View the Doubly Linked List.
3: Insert New Student
4: Delete a student whose name is specified by the user.
5: Delete all students that don't have health insurance.
input a choice:2
===== Double Linked list =====
    name6    lname6    12216    66      Math    1
    name4    lname4    12214    77      Math    0
    name7    lname7    22217    58      Math    1
    name5    lname5    22215    78      physics  0
    name3    lname3    22213    79      physics  1
    name1    lname1    22211    77      Math    1
=====

```

Figure 8: deleting a student using first and last name

```

***Doubly Linked LIST MENU ***
0: Return To Main Menu
1: Create a Doubly Linked List of all students.
2: View the Doubly Linked List.
3: Insert New Student
4: Delete a student whose name is specified by the user.
5: Delete all students that don't have health insurance.
input a choice:5
student with id [12214] with no health insurance was deleted from double linked list
student with id [22215] with no health insurance was deleted from double linked list

***Doubly Linked LIST MENU ***
0: Return To Main Menu
1: Create a Doubly Linked List of all students.
2: View the Doubly Linked List.
3: Insert New Student
4: Delete a student whose name is specified by the user.
5: Delete all students that don't have health insurance.
input a choice:2
===== Double Linked list =====
    name6    lname6    12216    66      Math    1
    name7    lname7    22217    58      Math    1
    name3    lname3    22213    79      physics  1
    name1    lname1    22211    77      Math    1
=====

```

Figure 9: delete uninsured students

## Linked Queue Options

```
***Linked Queue MENU ***
0: Return To Main Menu
1: Create a Linked Queue of all students from the highest ID to the lowest.
2: View the Linked Queue.
3: Enqueue.
4: Dequeue.
input a choice:1
Ordered Queue Created ...

***Linked Queue MENU ***
0: Return To Main Menu
1: Create a Linked Queue of all students from the highest ID to the lowest.
2: View the Linked Queue.
3: Enqueue.
4: Dequeue.
input a choice:2
===== Ordered Queue=====
    name7    lname7    22217    58    Math    1
    name5    lname5    22215    78    physics  0
    name3    lname3    22213    79    physics  1
    name1    lname1    22211    77    Math    1
    name6    lname6    12216    66    Math    1
    name4    lname4    12214    77    Math    0
    name2    lname2    12212    78    Math    1
=====
```

Figure 10: ordered queue created, sorted by id from highest to lowest



```

input a choice:4
  Dequeue of:      name5      lname5      22215      78      physics      0
Wrong Input (choose 0..4)

***Linked Queue MENU ***
0: Return To Main Menu
1: Create a Linked Queue of all students from the highest ID to the lowest.
2: View the Linked Queue.
3: Enqueue.
4: Dequeue.
input a choice:4
  Dequeue of:      name3      lname3      22213      79      physics      1
Wrong Input (choose 0..4)

***Linked Queue MENU ***
0: Return To Main Menu
1: Create a Linked Queue of all students from the highest ID to the lowest.
2: View the Linked Queue.
3: Enqueue.
4: Dequeue.
input a choice:4
  Dequeue of:      name1      lname1      22211      77      Math      1
Wrong Input (choose 0..4)

***Linked Queue MENU ***
0: Return To Main Menu
1: Create a Linked Queue of all students from the highest ID to the lowest.
2: View the Linked Queue.
3: Enqueue.
4: Dequeue.
input a choice:2
===== Ordered Queue=====
      name6      lname6      12216      66      Math      1
      name4      lname4      12214      77      Math      0
      name2      lname2      12212      78      Math      1
=====

```

Figure 11: sample dequeue of some students

```

1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:3
input first name:pushname
input last name:pushlname
input id:432
input GPA:66
input major:asdf
input insurance status (1 for insured) (0 for uninsured):1

***Doubly Linked Stack MENU ***
0: Return To Main Menu
1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:2
===== Ordered by GPA Double Linked Stack=====
  name3      lname3      22213      79      physics      1
  name2      lname2      12212      78      Math          1
  name5      lname5      22215      78      physics      0
  name1      lname1      22211      77      Math          1
  name6      lname6      12216      66      Math          1
  pushname   pushlname    432      66      asdf          1
  name7      lname7      22217      58      Math          1
=====

***Doubly Linked Stack MENU ***
0: Return To Main Menu
1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:

```

Figure 12: double linked stack menu, push example

```

2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:3
input first name:pushtop
input last name:pushtoplname
input id:43243
input GPA:88
input major:sadf
input insurance status (1 for insured) (0 for uninsured):1

***Doubly Linked Stack MENU ***
0: Return To Main Menu
1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:2
===== Ordered by GPA Double Linked Stack=====
  pushtop pushtoplname    43243    88      sadf      1
  name3      lname3      22213    79      physics    1
  name2      lname2      12212    78      Math       1
  name5      lname5      22215    78      physics    0
  name1      lname1      22211    77      Math       1
  name6      lname6      12216    66      Math       1
  pushname   pushlname    432     66      asdf       1
  name7      lname7      22217    58      Math       1
=====

***Doubly Linked Stack MENU ***

```

Figure 13: push top example

```

0: Return To Main Menu
1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:4
Dequeue of:   pushtop pushtoplname      43243      88      sadf      1
Wrong Input (choose 0..5)

***Doubly Linked Stack MENU ***
0: Return To Main Menu
1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:4
Dequeue of:   name3      lname3      22213      79      physics      1
Wrong Input (choose 0..5)

***Doubly Linked Stack MENU ***
0: Return To Main Menu
1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:4
Dequeue of:   name2      lname2      12212      78      Math      1
Wrong Input (choose 0..5)

***Doubly Linked Stack MENU ***
0: Return To Main Menu
1: Create a Doubly Linked Stack of all students from the highest GPA to the lowest.
2: View the Doubly Linked Stack.
3: Push.
4: Pop.
input a choice:2
===== Ordered by GPA Double Linked Stack=====
name5      lname5      22215      78      physics      0
name1      lname1      22211      77      Math      1
name6      lname6      12216      66      Math      1
pushname   pushlname      432      66      asdf      1
name7      lname7      22217      58      Math      1

```

Figure 14: pop examples

```

***BINARY TREE MENU ***
0: Return To Main Menu
1: Convert the array of Students into a balanced left justified binary tree.
2: View the sequence of nodes (student names) of the binary tree in pre-order.
3: View the sequence of nodes (student names) of the binary tree in post-order.
4: Generate the adjacency matrix of the binary tree.
input a choice:1
Binary Tree was Created ...

***BINARY TREE MENU ***
0: Return To Main Menu
1: Convert the array of Students into a balanced left justified binary tree.
2: View the sequence of nodes (student names) of the binary tree in pre-order.
3: View the sequence of nodes (student names) of the binary tree in post-order.
4: Generate the adjacency matrix of the binary tree.
input a choice:2
[name3 lname3]
[name2 lname2]
[name1 lname1]
[name5 lname5]
[name4 lname4]
[name7 lname7]
[name6 lname6]

***BINARY TREE MENU ***
0: Return To Main Menu
1: Convert the array of Students into a balanced left justified binary tree.
2: View the sequence of nodes (student names) of the binary tree in pre-order.
3: View the sequence of nodes (student names) of the binary tree in post-order.
4: Generate the adjacency matrix of the binary tree.
input a choice:

```

Figure 15: binary tree creation and preorder view (ordering is based on fname)

```

***BINARY TREE MENU ***
0: Return To Main Menu
1: Convert the array of Students into a balanced left justified binary tree.
2: View the sequence of nodes (student names) of the binary tree in pre-order.
3: View the sequence of nodes (student names) of the binary tree in post-order.
4: Generate the adjacency matrix of the binary tree.
input a choice:2
[name3 lname3]
[name2 lname2]
[name1 lname1]
[name5 lname5]
[name4 lname4]
[name7 lname7]
[name6 lname6]

***BINARY TREE MENU ***
0: Return To Main Menu
1: Convert the array of Students into a balanced left justified binary tree.
2: View the sequence of nodes (student names) of the binary tree in pre-order.
3: View the sequence of nodes (student names) of the binary tree in post-order.
4: Generate the adjacency matrix of the binary tree.
input a choice:3
[name1 lname1]
[name2 lname2]
[name4 lname4]
[name6 lname6]
[name7 lname7]
[name5 lname5]
[name3 lname3]

```

Figure 16: post order print of left balanced binary tree

```

[name4 lname4]
[name6 lname6]
[name7 lname7]
[name5 lname5]
[name3 lname3]

***BINARY TREE MENU ***
0: Return To Main Menu
1: Convert the array of Students into a balanced left justified binary tree.
2: View the sequence of nodes (student names) of the binary tree in pre-order.
3: View the sequence of nodes (student names) of the binary tree in post-order.
4: Generate the adjacency matrix of the binary tree.
input a choice:4

adjacency matrix generated from the binary tree:
0 0 0 0 0 0 0
1 0 0 0 0 0 0
0 1 0 1 0 0 0
0 0 0 0 1 1 0
0 0 0 0 0 0 1
0 0 0 0 0 0 0
0 0 0 0 0 0 0

***BINARY TREE MENU ***

```

Figure 17: adjacency matrix