```
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

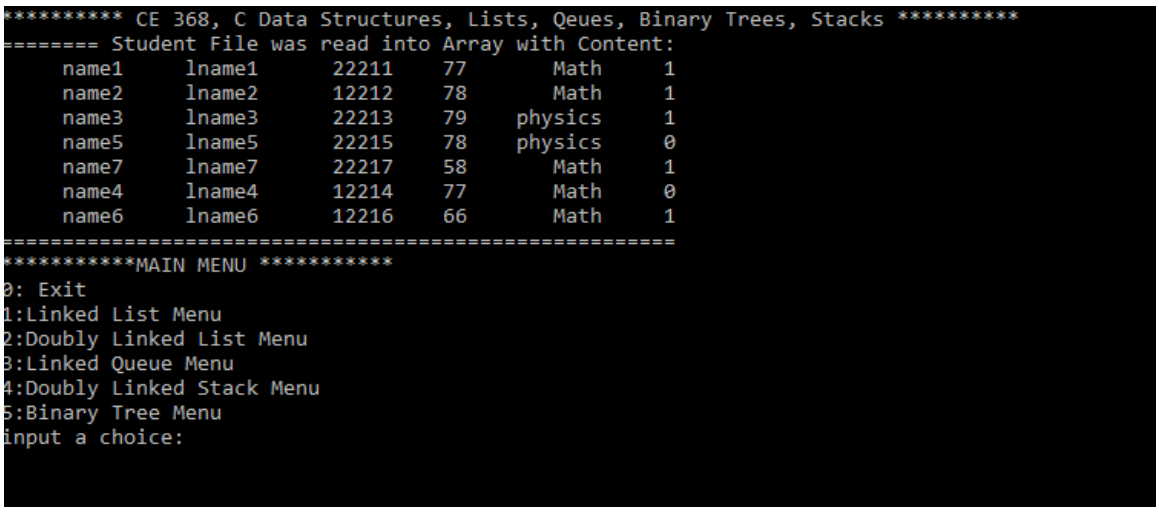

Figure 1: initial running the program, showing the content of the file read stored in array and presenting user options

LINKED LIST OPTIONS

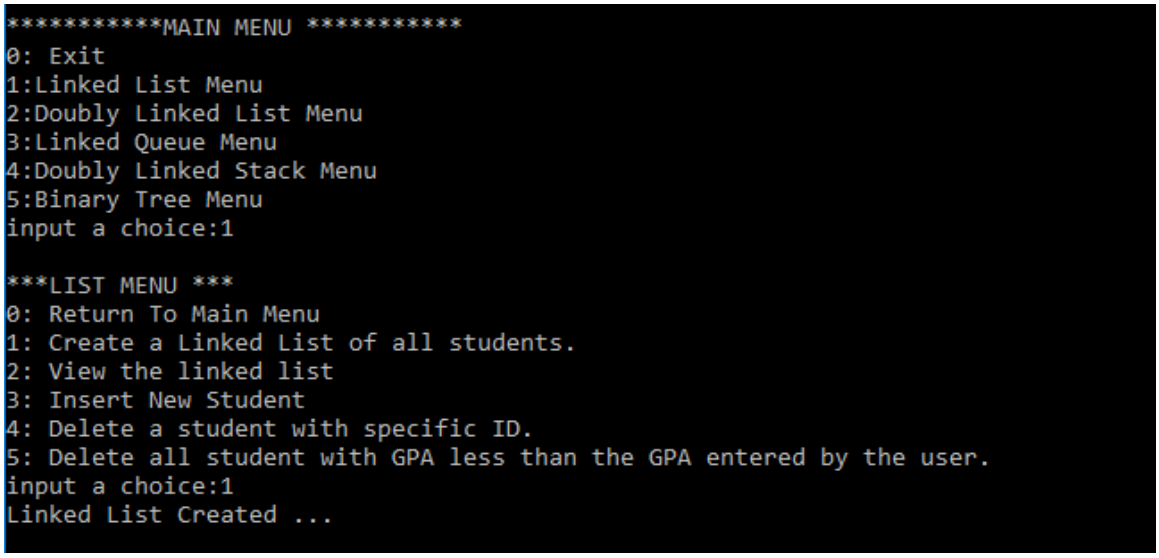

Figure 2: list menu and creation option

```
***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

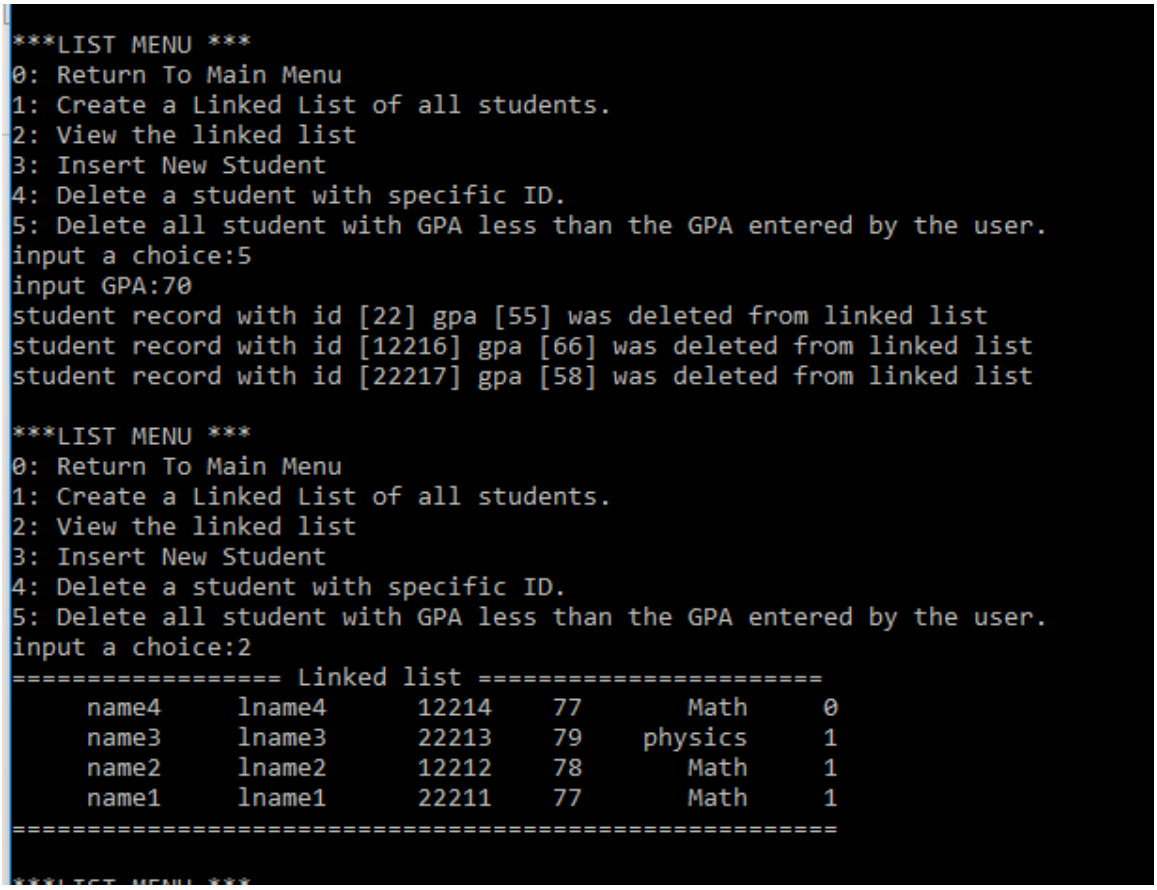

Figure 6: deleting students with GPA less than certain value, showing updated list

Double LINKED LIST OPTIONS

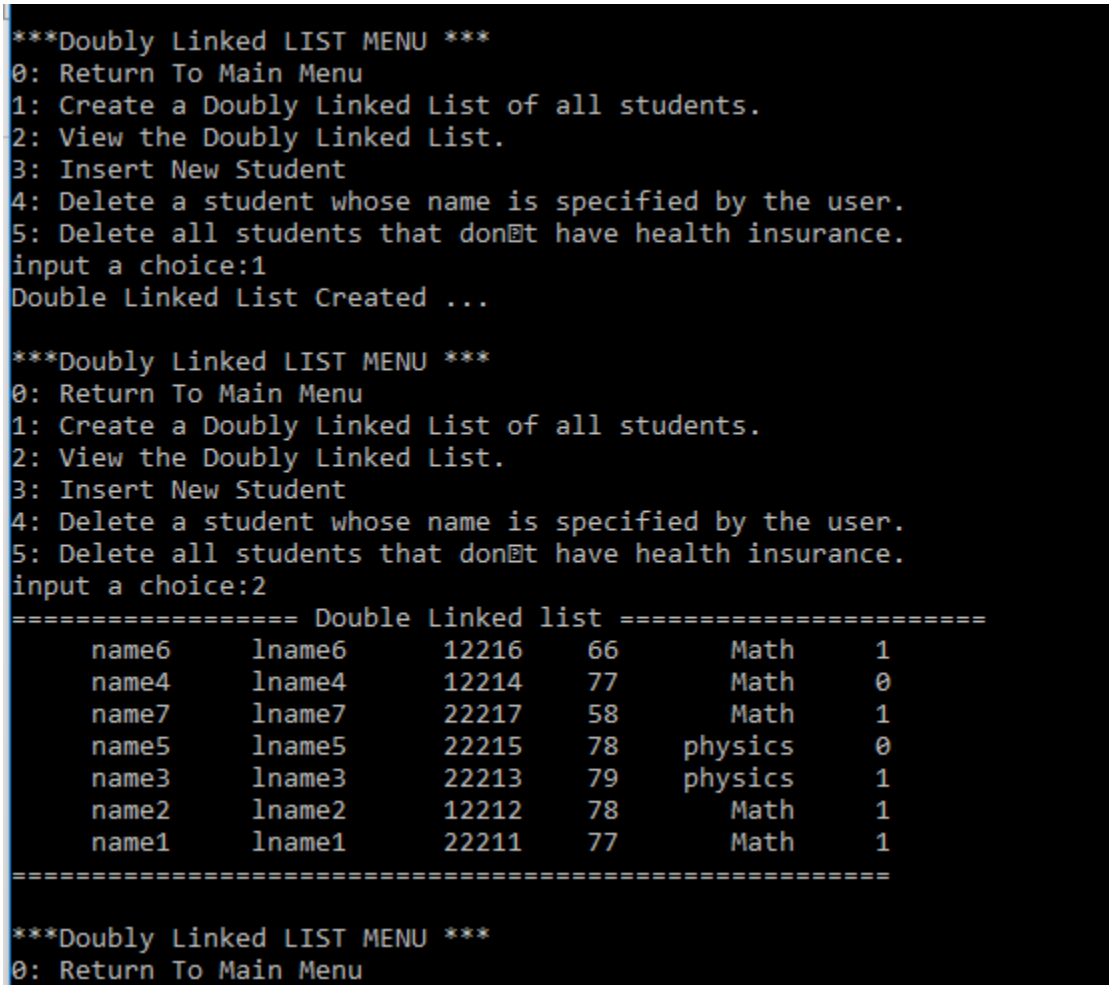

Figure 7: double linked list created from original student array

```
***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

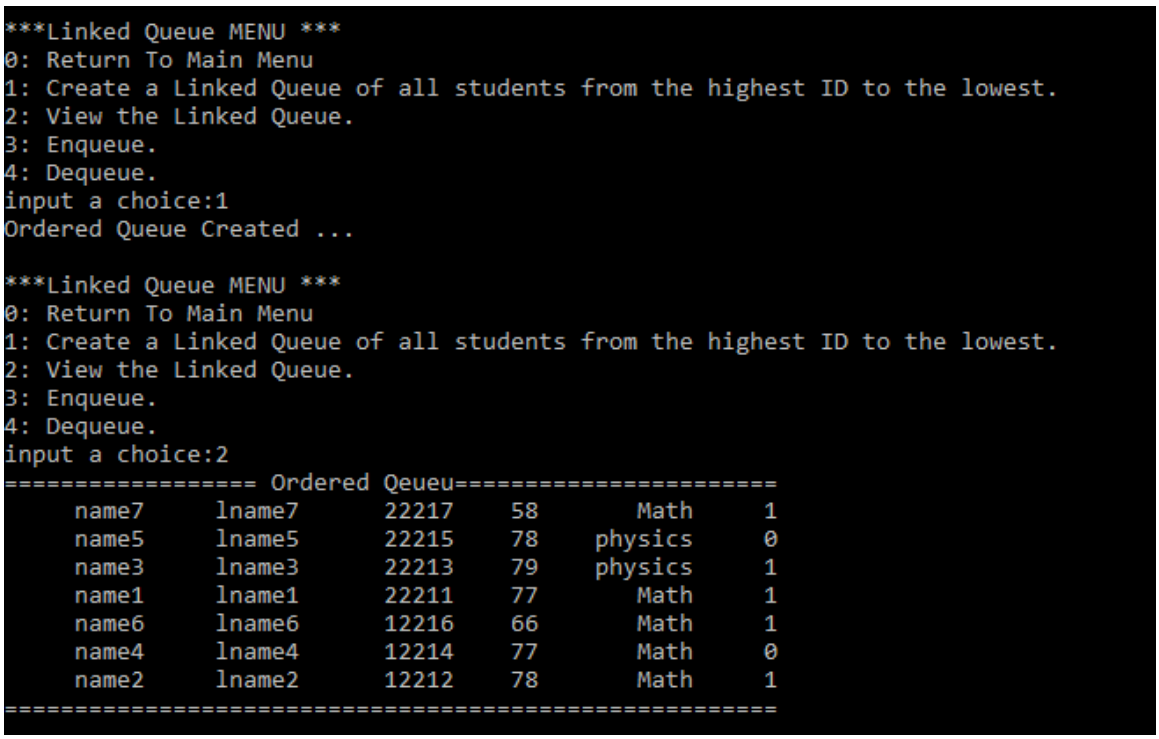

Figure 10: ordered queue created, sorted by id from highest to lowest

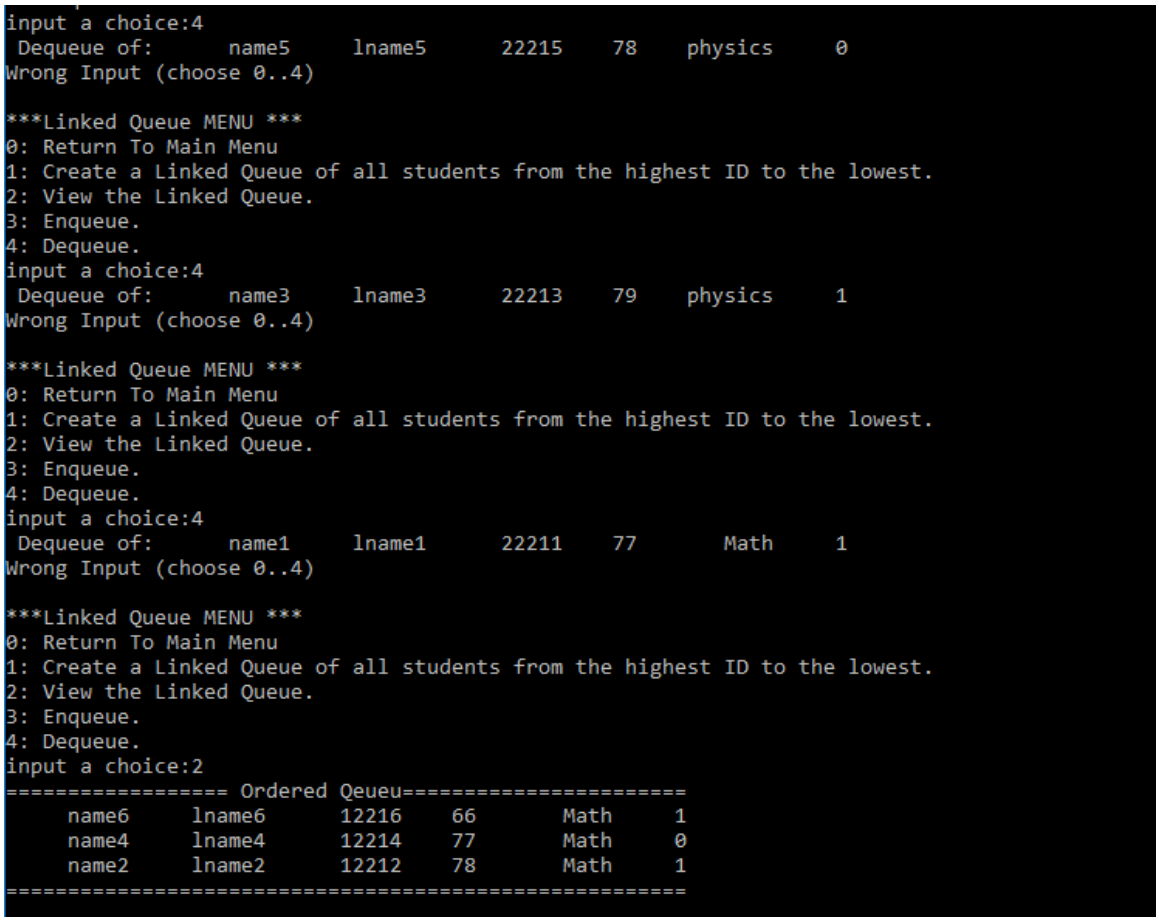

Figure 11: sample dequeue of some students

```
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

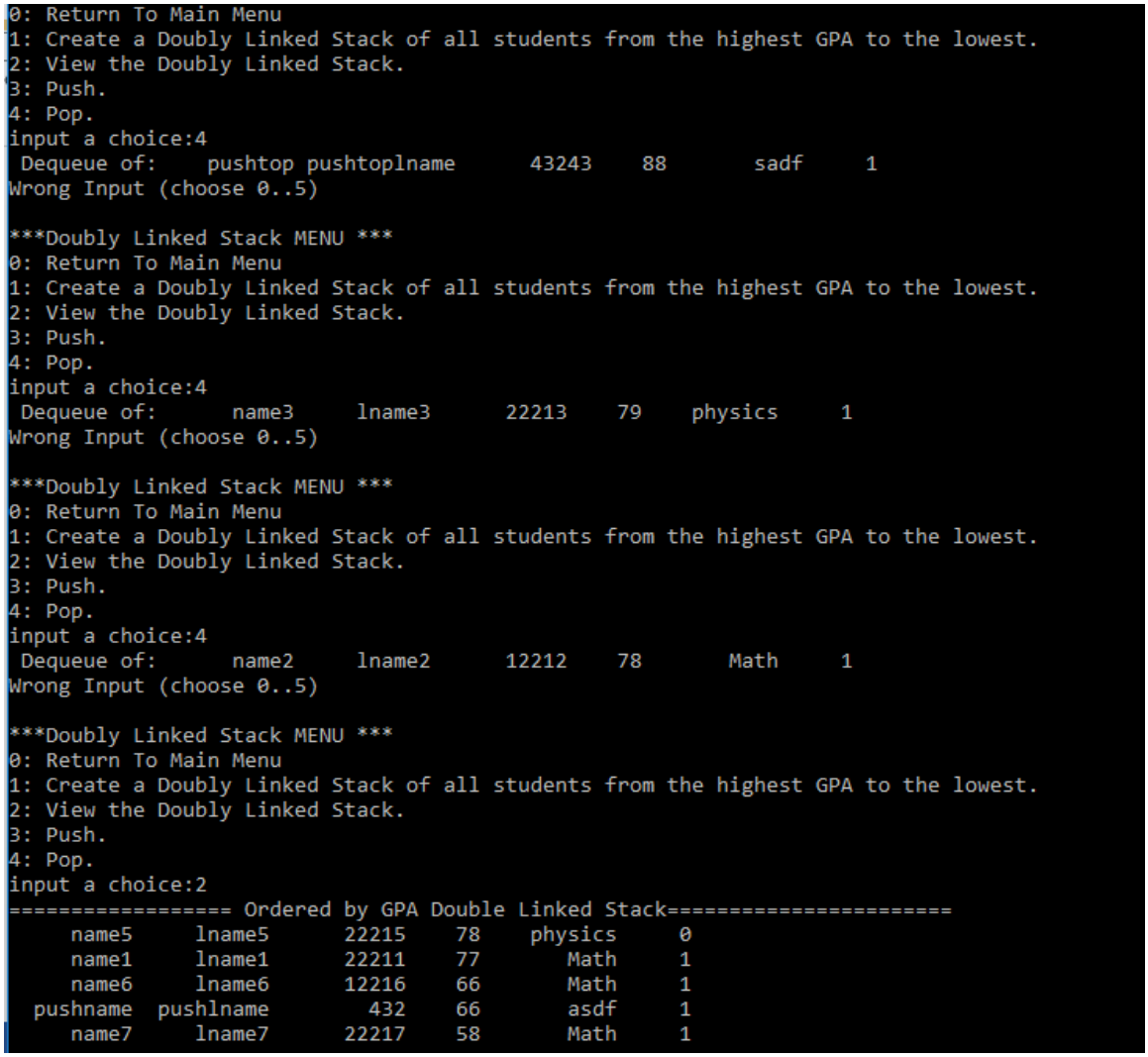

Figure 14: pop examples

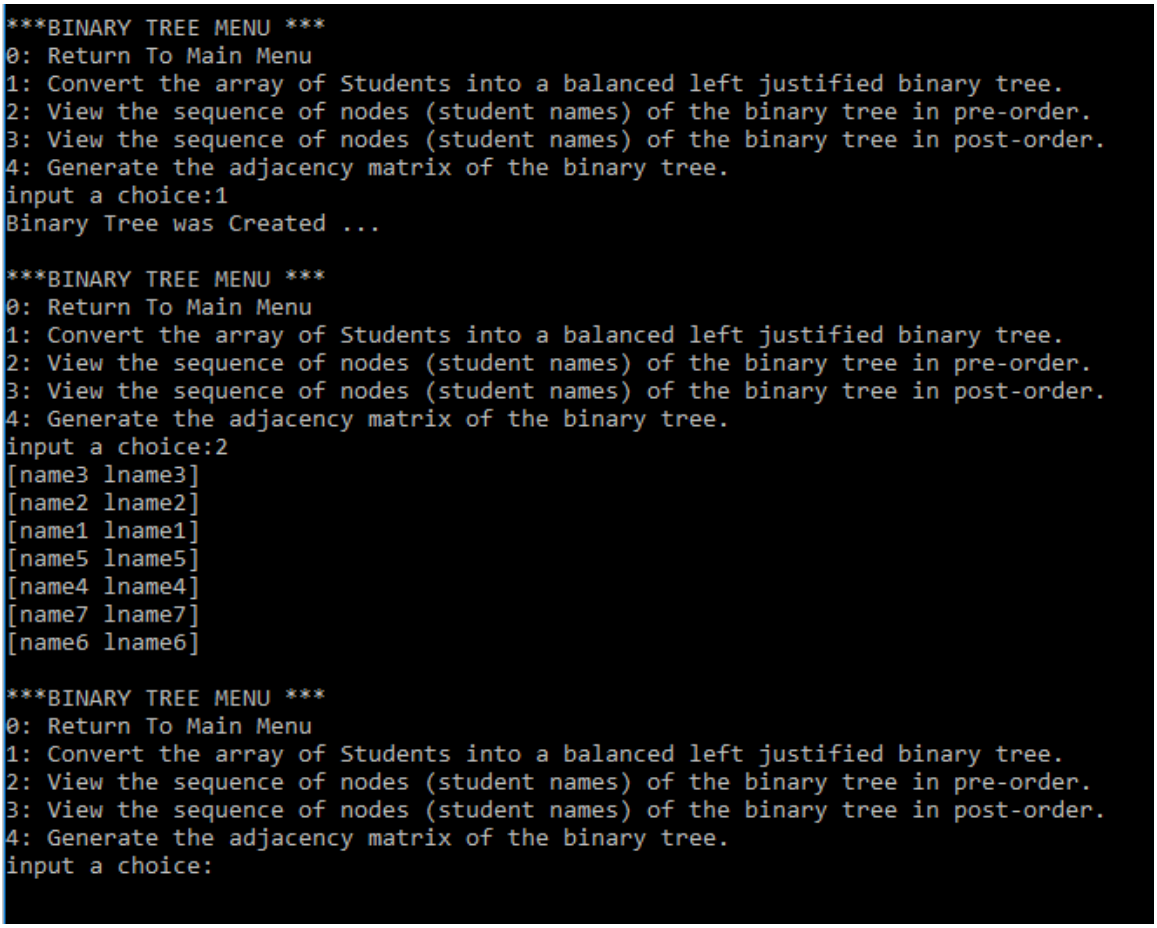

Figure 15: binary tree creation and preorder view (ordering is based on fname)

```
***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