

Problem Set 3 Report

Task 1

```
C:\Users\Rmuku\OneDrive - Swinburne Uni
All Nodes Without Sorting:
Prev      Current      Next
(<NULL>,  37.3,      20.6)
(37.3,     20.6,     138.9)
(20.6,     138.9,     70)
(138.9,     70,      55.9)
(70,       55.9,     25.9)
(55.9,     25.9,     144.4)
(25.9,     144.4,     66.9)
(144.4,     66.9,     112.6)
(66.9,     112.6,     106.7)
(112.6,     106.7,     134.2)
(106.7,     134.2,     129.5)
(134.2,     129.5,     <NULL>)

Prev      Current      Next
(<NULL>,  20.6,      25.9)
(20.6,     25.9,     37.3)
(25.9,     37.3,     55.9)
(37.3,     55.9,     66.9)
(55.9,     66.9,     70)
(66.9,     70,      106.7)
(70,       106.7,     112.6)
(106.7,     112.6,     129.5)
(112.6,     129.5,     134.2)
(129.5,     134.2,     138.9)
(134.2,     138.9,     144.4)
(138.9,     144.4,     <NULL>)
Press any key to continue . . .
```

Figure 1: Output of Task 1

For this program to work, two new functions were introduced. The first is the “sorting” function. This function takes in an array of any type depending on the data type given and sorts it ascendingly. The second function is the “makeDoublyLinkedList” function. This function takes in a double data type array and converts them into a doubly linked list. Then it will print the contents of the doubly linked list as shown in Figure 1. In the main, a double array is first created with data according to the question sheet given. It then uses the “makeDoublyLinkedList” by passing in the array previously mentioned. It then uses the “sorting” function to sort the array ascendingly. Next it uses the “makeDoublyLinkedList” function again with the sorted array.

Task 2

```
C:\Users\Rmuku\OneDrive - Swinburne U
All Nodes Without Sorting:
Prev      Current      Next
(<NULL>,   Emma,      Zack)
(Emma,     Zack,     Wade)
(Zack,     Wade,    Liam)
(Wade,     Liam,     Kyle)
(Liam,     Kyle,     Mason)
(Kyle,     Mason,    Fiona)
(Mason,     Fiona,   Sam)
(Fiona,     Sam,     Ava)
(Sam,      Ava,     Mike)
(Ava,      Mike,    Diana)
(Mike,     Diana,   Paul)
(Diana,    Paul,    Ryan)
(Paul,     Ryan,    Aidan)
(Ryan,     Aidan,   Beth)
(Aidan,    Beth,    Noel)
(Beth,     Noel,    Tina)
(Noel,     Tina,    Harry)
(Tina,     Harry,   Cyril)
(Harry,    Cyril,   Jean)
(Cyril,    Jean,    <NULL>)

All Nodes after Sorting:
Prev      Current      Next
(<NULL>,   Aidan,     Ava)
(Aidan,    Ava,      Beth)
(Ava,      Beth,     Cyril)
(Beth,     Cyril,    Diana)
(Cyril,    Diana,    Emma)
```

Figure 2: Output of Task 2 (pt1)

```
C:\Users\Rmuku\OneDrive - Swinburne Univers
(Noel, Tina, Harry)
(Tina, Harry, Cyril)
(Harry, Cyril, Jean)
(Cyril, Jean, <NULL>)

All Nodes after Sorting:
Prev      Current      Next
(<NULL>, Aidan, Ava)
(Aidan, Ava, Beth)
(Ava, Beth, Cyril)
(Beth, Cyril, Diana)
(Cyril, Diana, Emma)
(Diana, Emma, Fiona)
(Emma, Fiona, Harry)
(Fiona, Harry, Jean)
(Harry, Jean, Kyle)
(Jean, Kyle, Liam)
(Kyle, Liam, Mason)
(Liam, Mason, Mike)
(Mason, Mike, Noel)
(Mike, Noel, Paul)
(Noel, Paul, Ryan)
(Paul, Ryan, Sam)
(Ryan, Sam, Tina)
(Sam, Tina, Wade)
(Tina, Wade, Zack)
(Wade, Zack, <NULL>)
Press any key to continue . . .
```

Figure 3: Output of Task 2 (pt2)

Similar to Task 1, Task 2 also requires two additional functions to work. The first is the same “sorting” function as the function in Task 1. The second function is also a “makeDoublyLinkedList” function but this time it takes in a string array. The main function is the same as the main function in Task 1. The only difference is that the array created is a string array containing the names based on the question sheet.