

Assignment - 7

Rudra Chinhara, 202117

Write a program to create a linked list & perform operations such as insert, delete, update, reverse in the linked list.

- Code:

```
137 lines (129 sloc) 2.3 KB
1 #include <iostream>
2 using namespace std;
3
4 class node
5 {
6     public:
7     int data;
8     node *next;
9 };
10
11 node *start = NULL;
12
13 void insertNode(int info)
14 {
15     node *temp = new node;
16     temp->data = info;
17     temp->next = start;
18     start = temp;
19 }
20
21 void deleteNode()
22 {
23     node *temp, *prevnode;
24     if(start == NULL)
25     {
26         printf("List is empty\n");
27     }
28     else
29     {
30         temp = start;
31         while(temp->next != 0)
32         {
33             prevnode = temp;
34             temp = temp->next;
35         }
36         free(temp);
37         prevnode->next = 0;
38     }
39 }
40
41 void updateNode(int old, int newData)
42 {
43     int pos = 0, flag = 1;
44     node *temp;
45
46     if(start == NULL)
47     {
48         printf("Linked List is empty\n");
49     }
50     else
51     {
52         temp = start;
53         while(temp->next != NULL)
54         {
55             if(temp->data == old)
56             {
57                 temp->data = newData;
58                 printf("Node found at position %d, replaced with %d\n", old, pos, newData);
59                 flag = 0;
60             }
61             else
62             {
63                 pos++;
64                 temp = temp->next;
65             }
66         }
67         if(flag == 1)
68             printf("Node not found\n");
69     }
70 }
```

```

61         else
62         {
63             temp = temp->next;
64             pos++;
65         }
66     }
67 }
68 if(flag == 1)
69     printf("%d does not exist in the list\n", old);
70 }
71
72 void reverseLL()
73 {
74     node *t1, *t2, *temp;
75     t1 = t2 = NULL;
76
77     if(start == NULL)
78     {
79         printf("List is empty\n");
80     }
81     else
82     {
83         while(start != NULL)
84         {
85             //Reversing of points
86             t2 = start->next;
87             start->next = t1;
88             t1 = start;
89             start = t2;
90         }
91         start = t1;
92
93         //New head node
94         temp = start;
95         printf("Reversed linked list is: ");
96         while(temp != NULL)

```

```

97     {
98         printf("%d ", temp->data);
99         temp = temp->next;
100     }
101     printf("\n");
102 }
103 }
104
105 void traverseNode()
106 {
107     node *temp;
108     if (start == NULL)
109         printf("\nList is empty\n");
110     else
111     {
112         temp = start;
113         while(temp != NULL)
114         {
115             printf("%d ", temp->data);
116             temp = temp->next;
117         }
118         printf("\n");
119     }
120 }
121
122 int main()
123 {
124     insertNode(5);
125     insertNode(4);
126     insertNode(9);
127     insertNode(0);
128     insertNode(2);
129     traverseNode();
130     deleteNode();
131     traverseNode();
132     updateNode(9, 4);
133     traverseNode();
134     reverseLL();
135
136     return 0;
137 }

```

• Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
ist/" % g++ LinkedList.cpp -o LinkedList % % "/home/rudrachinhara/Documents/CUH/3rd Sem/Data Structure and Algorithms/DSA Assignments/Assignment-7/LinkedList/"LinkedList
rudrachinhara@dell-vostro:~/Documents/CUH/3rd Sem/Data Structure and Algorithms/DSA Assignments$ cd "/home/rudrachinhara/Documents/CUH/3rd Sem/Data Structure and Algorithms/DSA Assignments/Assignment-7/LinkedList/" % g++ LinkedList.cpp -o LinkedList % % "/home/rudrachinhara/Documents/CUH/3rd Sem/Data Structure and Algorithms/DSA Assignments/Assignment-7/LinkedList/"LinkedList
2 6 9 4 5
2 6 9 4
2 6 4 4
9 found at position 2, replaced with 4
2 6 4 4
Reversed linked list is: 4462
rudrachinhara@dell-vostro:~/Documents/CUH/3rd Sem/Data Structure and Algorithms/DSA Assignments/Assignment-7/LinkedList$

```