Discipline Core

Data Structures

CS4002

Test

Sheikh Muhammed Tadeeb (AU19B1014)

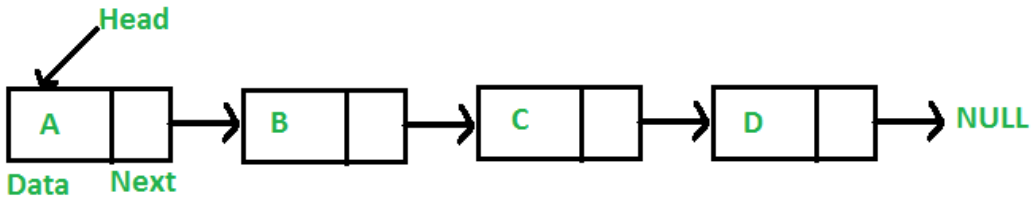
❖ Problem Statement:

Create a list of students for an institute database, write the C++ code with the appropriate data structure for storing the student's marks for six subjects and calculate the percentage of each student for all the subjects.

❖ Data-Type:

- **Linked-List:**

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:



In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

- **Why Linked-List?**

Time

Linked lists have most of their benefit when it comes to the insertion and deletion of nodes in the list. Unlike the dynamic array, insertion and deletion at any part of the list takes constant time.

However, unlike dynamic arrays, accessing the data in these nodes takes linear time because of the need to search through the entire list via pointers. It's also important to note that there is no way of optimizing search in linked lists. In the array, we could at least keep the array sorted. However, since we don't know how long the linked list is, there is no way of performing a binary search. Indexing - O(n), Insertion - O (1), Search - O(n). Deletion - O (1)

Space

Linked lists hold two main pieces of information (the value and pointer) per node. This means that the amount of data stored increases linearly with the number of nodes in the list. Therefore, the space complexity of the linked list is linear. Space - O(n).

❖ C++ Code: -

```
1. /*
2. Create a list of students for a institute database,
3. write the C++ code with the appropriate data structure
4. for storing the students marks for six subjects and
5. calculate the percentage of each student for all the subjects.
6. */
7.
8. //****************************** NOTE: The dataStructure I will use is
   LINKED-LIST.******************************
9.
10. #include <iostream>
11. using namespace std;
12.
13. // Here we are creating the linked-list structure.
14. class student_marks
15. {
16.       public:
17.             ~student_marks(){};
18.       public:
19.             string name;
20.             float math_marks;
21.             float science_marks;
22.             float english_marks;
23.             float biology_marks;
24.             float hindi_marks;
25.             float history_marks;
26.             float percentage;
27.             student_marks* next;
28.
29. };
30.
31. // This is the main class where we'll create all our functions
32. class marks : protected student_marks
33. {
34.             public:
35.             // This is the destructor.
```

```
36.             ~marks(){};
37.
38.         // Creating a variable node of type student_marks
39.             typedef student_marks node;
40.
41.             // Declaring the function for this class.
42.             node* create_list();
43.             void display(node* head);
44.             void search(node *head);
45.
46.             // Calling all the functions in this class
47.             void call(){
48.                     // Creating a pointer head to stode linked-list head
    address.
49.                     node* head;
50.                     head = create_list();
51.                     display(head);
52.                     search(head);
53.             }
54. };
55.
56. // This function is to create and insert the data in linked-list.
57. student_marks* marks :: create_list(){
58.                                                     int n;
59.                                                     float percent;
60.                                                     node *head,*p;
61.
        cout<<endl<<endl<<endl<<"\t\t\t\t##############  Student Record
    ##############";
62.
        cout<<endl<<endl<<endl<<endl<<"\tEnter the nos of students to be added
    = ";
63.                                                     cin>>n;
64.                                                     // Construction of linked-list
    nodes.
65.                                                     for (int i=0; i<n ; i++){
66.                                                             if (i==0){
67.                                                                     head = new node;
68.                                                                     p = head;
69.                                                             }
70.                                                             else{
71.                                                                     p->next = new
    node;
72.                                                                     p = p->next;
73.                                                             }
74.                                                             // Taking student marks
75.
        cout<<endl<<endl<<endl<<"\tEnter the details of "<<i+1<<" student";
```

```
76.
        cout<<endl<<endl<<"\tEnter students Name:- ";
77.                                                 cin>>p->name;
78.                                                 cout<<endl<<"\tEnter
   maths marks:- ";
79.                                                 cin>>p->math_marks;
80.                                                 cout<<endl<<"\tEnter
   science marks:- ";
81.                                                 cin>>p->science_marks;
82.                                                 cout<<endl<<"\tEnter
   english marks:- ";
83.                                                 cin>>p->english_marks;
84.                                                 cout<<endl<<"\tEnter
   biology marks:- ";
85.                                                 cin>>p->biology_marks;
86.                                                 cout<<endl<<"\tEnter
   hindi marks:- ";
87.                                                 cin>>p->hindi_marks;
88.                                                 cout<<endl<<"\tEnter
   history marks:- ";
89.                                                 cin>>p->history_marks;
90.                                                 //Calculating the
   percentage based on marks
91.                                                 percent = ( (p-
   >math_marks + p->science_marks + p->english_marks + p->biology_marks + p-
   >hindi_marks + p->history_marks) / 600 ) * 100;
92.                                     p->percentage = percent;
93.                                             }
94.                                             p->next = NULL;
95.                                             // Returning head address of
   linked-list for accessing the list.
96.                                             return head;
97. }
98.
99. // This function is for displaying the marks.
100.      void marks :: display(node* head){
101.                                             int count = 1;
102.                                             node *p;
103.                                             p = head;
104.
        cout<<endl<<endl<<endl<<endl<<"\t\t\t\t\t\t\t Student Record Info.";
105.                                             cout<<endl<<"\t\t
   ------------------------------------------------------------------------
   ------------------------------------------";
106.                                             cout<<endl<<endl<<"\t\t
   Sno.\t Name\t        Maths\t  Science\t  English\t  Biology\t  Hindi\t
   History\t  Percentage  ";
```

```
107.                                                    // Travesing over the
    whole list
108.                                                    while(p != NULL){
109.
        cout<<endl<<"\t\t
    ____________________________________________________________________
    ____________________________________________";
110.
        cout<<endl<<endl<<"\t\t  "<<count<<"\t  "<<p->name<<"\t  "<<p-
    >math_marks<<"\t   "<<p->science_marks<<"\t\t    "<<p->english_marks;
111.                                                            cout<<"\t
    "<<p->biology_marks<<"\t   "<<p->hindi_marks<<"\t   "<<p->history_marks<<"\t
    "<<p->percentage;
112.                                                            p = p->next;
113.                                                            count++;
114.                                                    }
115.                                                    cout<<endl<<"\t\t
    ____________________________________________________________________
    ____________________________________________";
116.                                            }
117.
118.      // This function is to search student details based on his/her name.
119.      void marks :: search(node* head){
120.            int count = 1;
121.            string sname;
122.            node *p;
123.            p = head;
124.            cout<<endl<<endl<<endl<<endl<<endl<<endl<<"_____________________
    ____________________________________________________________________
    ____________________________________________";
125.            cout<<endl<<endl<<"\t\t\t\t\t\t\t\tSearch";
126.            cout<<endl<<endl<<endl<<endl<<"\tEnter the Student Name whose
    details you wanna see:- ";
127.            cin>>sname;
128.            // Traversing over the linked-list
129.            while(p != NULL){
130.                  if(p->name == sname){
131.                        cout<<endl<<endl<<endl<<"\t\t\t\t\t\t\t"<<p-
    >name<<" Info.";
132.                        cout<<endl<<"\t\t
    ____________________________________________________________________
    ____________________________________________";
133.                        cout<<endl<<endl<<"\t\t  Sno.\t Name\t
    Maths\t  Science\t  English\t  Biology\t  Hindi\t  History\t  Percentage  ";
134.                        cout<<endl<<"\t\t
    ____________________________________________________________________
    ____________________________________________";
```

```
135.                            cout<<endl<<endl<<"\t\t  "<<count<<"\t  "<<p->name<<"\t  "<<p->math_marks<<"\t   "<<p->science_marks<<"\t\t    "<<p->english_marks;
136.                            cout<<"\t   "<<p->biology_marks<<"\t   "<<p->hindi_marks<<"\t   "<<p->history_marks<<"\t         "<<p->percentage;
137.                            cout<<endl<<"\t\t
    ____________________________________________________________________________________________________";
138.                            break;
139.                    }else{
140.                            p = p->next;
141.                    }
142.            }
143.    }
144.
145.    int main()
146.    {
147.            system("Color C0");
148.            marks obj;
149.            obj.call();     // Calling the call() function which contains all the functions.
150.            return 0;
151.    }
```

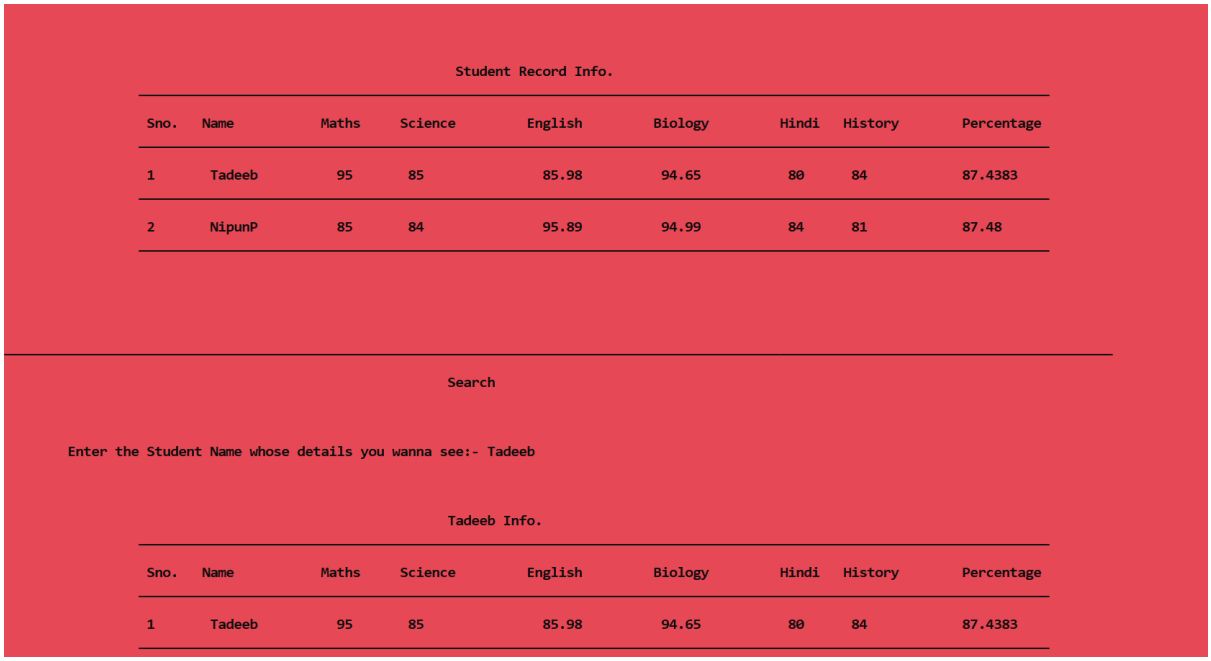
❖ C++ Output: -

```
                    ##############  Student Record  ##############

Enter the nos of students to be added = 2

Enter the details of 1 student
Enter students Name:- Tadeeb
Enter maths marks:- 95
Enter science marks:- 85
Enter english marks:- 85.98
Enter biology marks:- 94.65
Enter hindi marks:- 80
Enter history marks:- 84

Enter the details of 2 student
Enter students Name:- NipunP
Enter maths marks:- 85
Enter science marks:- 84
Enter english marks:- 95.89
Enter biology marks:- 94.99
Enter hindi marks:- 84
Enter history marks:- 81
```



Student Record Info.

Sno.	Name	Maths	Science	English	Biology	Hindi	History	Percentage
1	Tadeeb	95	85	85.98	94.65	80	84	87.4383
2	NipunP	85	84	95.89	94.99	84	81	87.48

Search

Enter the Student Name whose details you wanna see:- Tadeeb

Tadeeb Info.

Sno.	Name	Maths	Science	English	Biology	Hindi	History	Percentage
1	Tadeeb	95	85	85.98	94.65	80	84	87.4383

❖ Conclusion:

Well, I was successfully able to implement and do the operations on linked-list. From the assignment I got to know the working of pointers and how different memory allocations takes place. Working with linked list is better than traditional arrays as it saves us lot of memory which in case of arrays is either exhausted or remain untouched which is again a memory wastage.