Page:
Date: / /

Assignment 8

Class - SE IV
Batch - F4

Roll NO - 21030
DOS - 5/12/2020

Problem statement:-
Second yeam computer Engineering class Set A of Students like Vanilla Ice cream & Set B Students like butterscoch ice-cream Where C++ program to store two sets using linked list. Compute & display a) Set of Students who like both vanilla and butterscotch b) set of students who like either vanilla or butterscotch & not both c) Number of students who neither like Vanilla nor butterscotch.

Learning objectives-
1. To learn to write simple c++ program and execute it.
2. To implement singly Linked List in C++.

Learning Outcomes:-
1. Will be able to implement Linked list in C++.

S/W and H/W requirements:-
1. Open source C++ tools like GH.
2. Open source IDE Eclipse.
3. Windows 10 64 bit 8 GB RAM.

Page:
Date: / /

Theory:-

* Linked List -:

A linked list has a data structure which is linear. The elements in a linked list are not stored at continous memory locations. The basic structure of any linked list is a Node which contains minimum 2 fields which stores the address of next node. A single Node has always NULL in its next field. The first node in a Linked list is always pointed by a head pointer.

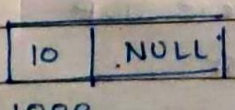
Generally there are four types of Linked List based on their Node structure.

1. Singly Linked List.
2. Doubly Linked List
3. Circular Linked List
4. Double Circular Linked List.

* Singly Linked List -:

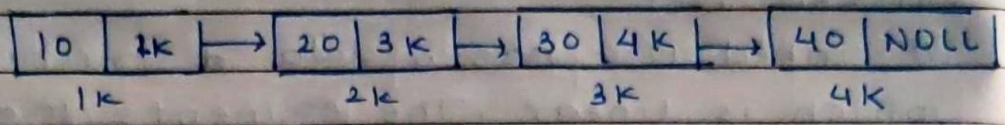
In Singly Linked List the Node contains only two data fields. i.e. data and next pointer. The data section stores the value and next field stores address of the next Node.

The structure of Node for SLL is



Page :
Date : / /

The structure of a Singly Linked list :-



* Pseudocode :-

* ADT structure for class student :-
  char name[30] // Store name of student
  int roll // Store roll No of student
  student *next // store add of next student

  getdata() // Assign value of all data members

* ADT structure for class List
  student *head // Store add of first Node.
  create() // create list of student.
  append() // Append student in list.
  show() // Display info of students
  show_both() // Display list of student who like both ice-cream
  show_either() // Display student who like either ice-cream
  show_neither() // Display studen who don't lik both.

* procedure create()
  While user no enters STOP
    create student
    input name, roll no,
    call append() passing it student
  END While

END.

* Procedure show()
  Node * p = head
  While (p is not NULL)
    Display name & roll No
    p = p -> next
  END While.
END

* procedure show_both()
  Node * p = Head of Vanilla list
  Node * q = Head of Butterscotch list.
  While (p is not NULL)
    While (q in not null)
      if (p -> data == q -> data)
        Display p -> data
      q = q -> next
    END While
    P = P -> next
  END While
END

* Procedure show_either
  Node * P = Head of Vanilla list
  Node * q = Head of butterscotch list
  While (P is not NULL)
    While (q is not NULL)
      if (P -> data != q -> data)
        show q -> data ;
      q = q -> next ;

```
        END While
        P= P->next.
    END While
END

procedure count
    int count =0
    Node *P= Head
    While(P is not NULL)
        count ++;
        P=P->next
    END While
    Return count
END
```

Complexity

function	Time complexity	Space Complexity
create()	$O(n)$	$O(n)$
show()	$O(n)$	$O(1)$
show_both()	$O(n^2)$	$O(n)$
count	$O(1)$	$O(1)$
show_either	$O(n^2)$	$O(n)$

Page:
Date: / /

* Test Cases -:

NO.	Description	Input	Expected o/p	Actual o/p	Result
1.	Menue:	ch -1			
	1. Create.	Name - Abc	created	created	
	2. Both	Van - y	List	List	
	3. Either	Butt -y			Pass
	4. CNT of None	Roll No -30			
		Name - xyz			
		Van -y			
		Butt -n			
		Name - STOP			
		Roll NO -40			
2	Menue	Ch - 2	Name - Abc	Name - Abc	
	1. Create		Van - y	Van - y	
	2. Both		Butt - y	Butt - y	Pass
	3. Either		Roll NO - 30	Roll NO - 30	
	4. CNT of None				

* Conclusion -:
We learnt to implement Linked List in Ch.

```cpp
#include <iostream>
#include <stdlib.h>
#include <string.h>
using namespace std;
class Node
{
    public:
    char name[20];
    int roll;
    Node *next;
    Node()
    {
        cout<<"\nEnter name ";
        cin.getline(name,18);
        cout<<"\nEnter roll no ";
        cin>>roll;
        next=NULL;
    }
    Node(char *n,int r)
    {
        strcpy(name,n);
        roll=r;
        next=NULL;
    }
};
class SLL
{

    public:
        Node *head;
        SLL()
        {
```

```
        }
        void create(SLL *,SLL *);
        void show();
        int count();
        void addatend(char*,int);
        void both(SLL *);
};
int main()
{
    SLL V,B,N;
    int ch;

    while(1)
    {
        cout<<"\nMenu";
        cout<<"\n1.Insert";
        cout<<"\n2.Display Both";
        cout<<"\n3.Display Who like either only one not both";
        cout<<"\n4.Display Number who does Not like any Ice cream";
        cout<<"\n0.Exit";
        cout<<"\n\nChoice : ";
        cin>>ch;
        cin.ignore(1);
        switch(ch)
        {
            case 1:
            {
                N.create(&V,&B);
                break;
            }
            case 2:
            {
```

Report ...

```
67              V.both(&B);
68              break;
69          }
70          case 3:
71          {
72              cout<<"\n\nThe students liking only vanilla are \n";
73              V.show();
74              cout<<"\n\nThe students liking only Butterscotch are \n";
75              B.show();
76               break;
77          }
78          case 4:
79              {
80                  cout<<"\n\nNumber of students who like neither vanilla nor butterscotch are : ";
81                  cout<<N.count();
82                  break;
83              }
84          case 5:
85              {
86                  cout<<"\n\nThe students not linking any ice cream are\n";
87                  N.show();
88                  break;
89
90              }
91          case 0:
92              {
93                  exit(1);
94              }
95          default:
96              {
97                  cout<<"\nInvailid Input ";
98
```

```
            }
        }
    }
    return 0;
}
void SLL::both(SLL *B)
{
    Node *p,*q;
    p=B->head;
    q=head;
    if(head==NULL||p==NULL)
    {
        cout<<"\nNo One like Both Icecream";
        return;
    }
    SLL s3;
    while(p!=NULL)
    {
        while(q!=NULL)
        {
            if(p->roll==q->roll)
            {
                s3.addatend(p->name,p->roll);
                break;
            }
            q=q->next;
        }
        p=p->next;
    }
    s3.show();
}
int SLL::count()
```

Report ...

```
{
    Node * p=head;
    int cnt=0;
    while(p!=NULL)
    {
        cnt++;
        p=p->next;
    }
    return cnt;
}
void SLL::create(SLL *S1,SLL *S2)
{
    if(head!=NULL)
    {
        cout<<"\nList already created";
        return;
    }
    cout<<"\nEnter  continue data\nIf you want ot stop enter STOP\n";
    char nam[10];
    int x;
    while(1)
    {
        int flag=0;
        cout<<"\nEnter name : ";
        cin.getline(nam,18);
        if(strcmp(nam,"STOP")==0){break;}
        cout<<"\nEnter roll no : ";
        cin>>x;
        cin.ignore();
        cout<<"\nIf Like vanilla Enter y or n \nVanilla : ";
        char ch;
        cin>>ch;
```

```
164         {
165             flag=1;
166             S1->addatend(nam,x);
167         }
168         cout<<"\nIf Like butterscotch Enter y or n \nButterscotch : ";
169         cin>>ch;
170         cin.ignore();
171         if(ch=='y')
172         {
173             flag=1;
174             S2->addatend(nam,x);
175         }
176         if(flag==0)
177         {
178             addatend(nam,x);
179         }
180
181     }
182 }
183 void SLL::show()
184 {
185     Node *p;
186     p=head;
187     if(head==NULL)
188     {
189         cout<<"\nEmpty list";
190         return;
191     }
192     while(p!=NULL)
193     {
194         cout<<"\nName : "<<p->name<<"\nRoll no : "<<p->roll;
195         p=p->next;
```

```
{
    Node *p;
    p=head;
    if(head==NULL)
    {
        cout<<"\nEmpty list";
        return;
    }
    while(p!=NULL)
    {
        cout<<"\nName : "<<p->name<<"\nRoll no : "<<p->roll;
        p=p->next;
    }
}
void SLL::addatend(char* n,int x)
{
    Node* q=NULL;
    q=new Node(n,x);
    if(head==NULL)
    {
        head=q;
    }
    else
    {
        Node*p=head;
        while(p->next!=NULL)
        {
            p=p->next;
        }
        p->next=q;
    }
}
```

F:\Ganesh\DEV CPP\program\FDS8.exe

```
Menu
1.Insert
2.Display Both
3.Display Who like either only one not both
4.Display Number who does Not like any Ice cream
0.Exit

Choice : 1

Enter  continue data
If you want ot stop enter STOP

Enter name : Ganesh

Enter roll no : 30

If Like vanilla Enter y or n
Vanilla : y

If Like butterscotch Enter y or n
Butterscotch : y

Enter name : Harshal

Enter roll no : 24

If Like vanilla Enter y or n
Vanilla : n

If Like butterscotch Enter y or n
Butterscotch : n

Enter name : Amit

Enter roll no : 45

If Like vanilla Enter y or n
Vanilla : y

If Like butterscotch Enter y or n
Butterscotch : n

Enter name : Rajat

Enter roll no : 59

If Like vanilla Enter y or n
Vanilla : n

If Like butterscotch Enter y or n
Butterscotch : y

Enter name : STOP

Menu
```

F:\Ganesh\DEV CPP\program\FDS8.exe

```
2.Display Both
3.Display Who like either only one not both
4.Display Number who does Not like any Ice cream
0.Exit

Choice : 2


The students liking both ice cream are

Name : Ganesh
Roll no : 30
Menu
1.Insert
2.Display Both
3.Display Who like either only one not both
4.Display Number who does Not like any Ice cream
0.Exit

Choice : 3


The students liking only vanilla are

Name : Ganesh
Roll no : 30
Name : Amit
Roll no : 45

The students liking only Butterscotch are

Name : Ganesh
Roll no : 30
Name : Rajat
Roll no : 59
Menu
1.Insert
2.Display Both
3.Display Who like either only one not both
4.Display Number who does Not like any Ice cream
0.Exit

Choice : 4


Number of students who like neither vanilla nor butterscotch are : 1
Menu
1.Insert
2.Display Both
3.Display Who like either only one not both
4.Display Number who does Not like any Ice cream
0.Exit

Choice : 0

--------------------------------
```