**Assignment 5:**

Department of Computer Engineering has student's club named 'Pinnacle Club'. Students

of second, third and final year of department can be granted membership on request.

Similarly one may cancel the membership of club. First node is reserved for president of

club and last node is reserved for secretary of club. Write C++ program to maintain club

member‘s information using singly linked list. Store student PRN and Name. Write

functions to:

a) Add and delete the members as well as president or even secretary.

b) Compute total number of members of club

c) Display members

d) Two linked lists exists for two divisions. Concatenate two lists.

|  |
| --- |
| /\*-----------------------------------------------------  Assignment 5:  Department of Computer Engineering has student's club named 'Pinnacle Club'. Students  of second, third and final year of department can be granted membership on request.  Similarly one may cancel the membership of club. First node is reserved for president of  club and last node is reserved for secretary of club. Write C++ program to maintain club  member‘s information using singly linked list. Store student PRN and Name. Write  functions to:  a) Add and delete the members as well as president or even secretary.  b) Compute total number of members of club  c) Display members  d) Two linked lists exists for two divisions. Concatenate two lists.  -----------------------------------------------------\*/  // Solution -  #include <iostream>  using namespace std;  class sll  {  struct node  {  int prn;  char name[10];  node \*next;  } \* start;  public:  sll()  {  start = NULL;  }  void create();  void display();  void insert\_beginning();  void insert\_end();  void insert\_mid();  void del\_beginning();  void del\_end();  void del\_mid();  int compute();  void concatenate(sll obj2);  };  void sll::create()  {  node \*temp;  node \*curr;  int ans;  do  {  temp = new node;  temp->next = NULL;  cout << "Enter PRN number :" << endl;  cin >> temp->prn;  cout << "Enter name: " << endl;  cin >> temp->name;  if (start == NULL)  {  start = temp;  curr = temp;  }  else  {  curr->next = temp;  curr = temp;  }  cout << "Do you want to add a new node ? 1 for yes" << endl;  cin >> ans;  } while (ans == 1);  }  void sll::display()  {  node \*t;  if (start == NULL)  {  cout << "Club is empty" << endl;  return;  }  else  {  t = start;  while (t != NULL)  {  cout << t->prn << " " << t->name << " ";  t = t->next;  }  }  }  void sll::insert\_beginning()  {  node \*temp;  temp = new node;  temp->next = NULL;  cout << "Enter PRN number: " << endl;  cin >> temp->prn;  cout << "Enter name: " << endl;  cin >> temp->name;  if (start == NULL)  {  start = temp;  }  else  {  temp->next = start;  start = temp;  }  }  void sll::insert\_end()  {  node \*temp, \*last;  temp = new node;  temp->next = NULL;  cout << "Enter PRN number: " << endl;  cin >> temp->prn;  cout << "Enter name: " << endl;  cin >> temp->name;  if (start == NULL)  {  start = temp;  }  else  {  last = start;  while (last->next != NULL)  {  last = last->next;  }  last->next = temp;  }  }  void sll::insert\_mid()  {  node \*temp;  node \*curr;  int no;  cout << "\nEnter PRN after you want to insert: " << endl;  cin >> no;  temp = new node;  temp->next = NULL;  cout << "Enter PRN number: " << endl;  cin >> temp->prn;  cout << "Enter name: " << endl;  cin >> temp->name;  curr = start;  for (int i = 1; i < no; i++)  {  curr = curr->next;  }  temp->next = curr->next;  curr->next = temp;  }  void sll::del\_beginning()  {  node \*temp;  if (start == NULL)  {  cout << "\nClub is empty" << endl;  }  else  {  temp = start;  start = start->next;  temp->next = NULL;  delete temp;  cout << "\nfirst element deleted" << endl;  }  }  void sll::del\_end()  {  node \*temp, \*prev;  if (start == NULL)  {  cout << "\nClub is empty" << endl;  }  else  {  while (temp->next != NULL)  {  prev = temp;  temp = temp->next;  }  prev->next = NULL;  delete temp;  cout << "\nlast element deleted." << endl;  ;  }  }  void sll::del\_mid()  {  node \*temp;  node \*curr;  int no;  cout << "\nEnter PRN after you want to delete: " << endl;  cin >> no;  temp = new node;  temp->next = NULL;  curr = start;  for (int i = 1; i < no; i++)  {  curr = curr->next;  }  temp->next = curr->next;  curr->next = temp;  delete temp;  cout << "\nelement deleted" << endl;  }  int sll::compute()  {  node \*temp;  int count = 0;  if (start == NULL)  {  cout << "\nClub is empty" << endl;  }  temp = start;  while (temp != NULL)  {  count++;  temp = temp->next;  }  return count;  }  void sll::concatenate(sll obj2)  {  node \*temp, \*last;  last = obj2.start;  if (last == NULL)  {  cout << "\nList 2 is empty" << endl;  return;  }  temp = start;  while (temp->next != NULL)  {  temp = temp->next;  }  temp->next = last;  cout << "\nAfter concatenation: ";  }  int main()  {  sll obj;  int ch;  do  {  cout << "\n1. create\n2.Insert at beginning\n3.Insert at end\n4.insert after position\n5.Display list\n6.Delete first element\n7.Delete last element\n8.Delete Member\n9.Find total No. of members\n10.Concatenate lists\n0. Exit\nEnter your choice: ";  cin >> ch;  switch (ch)  {  case 1:  obj.create();  obj.display();  break;  case 2:  obj.insert\_beginning();  obj.display();  break;  case 3:  obj.insert\_end();  obj.display();  break;  case 4:  obj.insert\_mid();  obj.display();  break;  case 5:  obj.display();  break;  case 6:  obj.del\_beginning();  obj.display();  break;  case 7:  obj.del\_end();  obj.display();  break;  case 8:  obj.del\_mid();  obj.display();  break;  case 9:  obj.compute();  obj.display();  break;  case 10:  {  sll obj2;  cout << "\nList 1: " << endl;  obj.create();  cout << "\nList 2: " << endl;  obj2.create();  obj.concatenate(obj2);  obj.display();  break;  }  }  } while (ch != 0);  cout << "\nEnd of program" << endl;  return 0;  } |