```
#include<iostream>
using namespace std;

class Node{
	public:
		int rollno;
		string name;
		long long int prn;
		bool placed;
		Node *next; //address of next node
};

class sll{
	Node *head; //address of node
	public:
		sll(){
			head = NULL;
		}
		void create();
		void display();
		void insert();
		void reverse();
		void search();
		void update();
		void sort();
		void del();
};

void sll::create(){
	char ch;
```

```
Node *temp = NULL; // pointer for a particular node
Node *p = NULL; //for connectivity of two nodes
temp = new(Node); //new node is created
do{
    if(head == NULL){ //if the head is not declared any temp node.
        cout << "Enter Roll no: ";
        cin >> temp->rollno;
        cout << "Enter name of student: ";
        cin >> temp->name;
        cout << "Enter PRN: ";
        cin >> temp->prn;
        cout << "Enter is the student is placed:(0 - false/ 1 - true): " ;
        cin >> temp->placed;
        cout << "Added Successfully!\n\n";

        head = temp;
        p = temp;
        temp->next = NULL;
    }
    else{ //head is already declared so creating other node and pointing address from the previous node to this one.
        temp = new(Node);
        cout << "Enter Roll no: ";
        cin >> temp->rollno;
        cout << "Enter name of student: ";
        cin >> temp->name;
        cout << "Enter PRN: ";
        cin >> temp->prn;
        cout << "Enter is the student is placed:(0 - false/ 1 - true): " ;
        cin >> temp->placed;
        cout << "Added Successfully!\n\n";
```

```
                    p->next = temp; //pointing address from first node to next node (here temp is holding the next node).
                    p = temp;
                    temp->next = NULL;
                }
                cout << "Do you want to continue? (y): ";
                cin >> ch;
        }while(ch=='y');
}

void sll::display(){
        Node *p;
        p = head;
        if(head == NULL){
                cout << "Linked list is empty!\n\n";
        }
        else{
                while(p!=NULL){
                        cout << "Roll no: " << p->rollno << endl;
                        cout << "Name: " << p->name << endl;
                        cout << "PRN: " << p->prn << endl;
                        cout << "Placed: " << p->placed << "\n\n";
                        p = p->next;
                }
        }
}

void sll::insert(){
        Node *temp = NULL; // pointer for a particular node
        Node *p = NULL; //for connectivity of two nodes
```

```
int ch = 0;
while(ch!=5){
	cout << "Enter:\n1 - Insert at Begining\n2 - Insert at end\n3 - Insert at Specific Position\n4 - Insert in between\n5 - Exit\nEnter: ";
	cin >> ch;
	if(ch == 5){
		break;
	}
	switch (ch){
		case 1:
			temp = NULL;
			temp = new(Node);
			cout << "Enter Roll no: ";
			cin >> temp->rollno;
			cout << "Enter name of student: ";
			cin >> temp->name;
			cout << "Enter PRN: ";
			cin >> temp->prn;
			cout << "Enter is the student is placed:(0 - false/ 1 - true): " ;
			cin >> temp->placed;

			temp->next = NULL;
			temp->next = head;
			head = temp;
			cout << "Node Added at Begining Successfully!\n\n";
			display();
			break;
		case 2:
			temp = NULL;
			temp = new(Node);
			cout << "Enter Roll no: ";
```

```
            cin >> temp->rollno;
            cout << "Enter name of student: ";
            cin >> temp->name;
            cout << "Enter PRN: ";
            cin >> temp->prn;
            cout << "Enter is the student is placed:(0 - false/ 1 - true): " ;
            cin >> temp->placed;

            temp->next = NULL;
            p = head;
            while(p->next!=NULL){
                  p = p->next;
            }
            p->next = temp;
            cout << "Node Added at Last Successfully!\n\n";
            display();
            break;
case 3:
            int x;
            cout << "Enter at what position you want to enter: ";
            cin >> x;
            temp = NULL;
            p = head;
            for(int i=0; i<x-1;i++){
                  if(p->next == NULL){
                        p = NULL;
                        cout << "Please enter a valid position!\n\n";
                        break;
                  }
                  p=p->next;
```

```
                }
                temp = new(Node);
                cout << "Enter Roll no: ";
                cin >> temp->rollno;
                cout << "Enter name of student: ";
                cin >> temp->name;
                cout << "Enter PRN: ";
                cin >> temp->prn;
                cout << "Enter is the student is placed:(0 - false/ 1 - true): " ;
                cin >> temp->placed;

                temp->next = p->next;
                p->next = temp;
                cout << "Node Added at " << x << " position
Successfully!\n\n";
                display();
                break;
        case 4:
                int n,f;
                temp = NULL;
                p = head;
                cout << "Enter value after which the roll no to be inserted: ";
                cin >> n;
                while(p->next != NULL){
                        if(p->rollno == n){
                                f = 1;
                                break;
                        }
                        p = p->next;
                }
                temp = new(Node);
```

```
                cout << "Enter roll no to inserted: ";
                cin >> temp->rollno;
                cout << "Enter name of student: ";
                cin >> temp->name;
                cout << "Enter PRN: ";
                cin >> temp->prn;
                cout << "Enter is the student is placed:(0 - false/ 1 - true): " ;
                cin >> temp->placed;

                temp->next = NULL;
                if(f==1){
                        if(p->next!=NULL){
                                temp->next = p->next;
                                p->next = temp;
                        }else{
                                p->next=temp;
                        }
                }else{
                        cout << "Element Not Found\n\n";
                }
                cout << "Linked List: \n";
                display();
                break;

        default:
                cout << "Enter valid choice!\n\n";
                break;
    }
  }
}
```

```
void sll::reverse(){
	Node *p;
	Node *q = NULL;
	p = head;
	Node *r;
	r = p->next;
	while(p!=NULL){
		p->next = q;
		q = p;
		p = r;
		if(p != NULL) r=r->next;
	}
	head = q;
	cout << "The reversed linked list is: \n";
	display();
}

void sll::search(){
	Node *p = NULL;
	int no;
	bool flag = false;
	p=head;
	cout <<"Enter the roll no. to search: ";
	cin >> no;
	while(p!=NULL){
		if(p->rollno == no){
			flag = true;
			break;
		}
```

```
            p = p->next;
        }
        if(flag){
            cout << "Roll no.: "<< no <<" is present!: \n\nDetails: \n";
            cout << "Roll no: " << p->rollno << endl;
            cout << "Name: " << p->name << endl;
            cout << "PRN: " << p->prn << endl;
            cout << "Placed: " << p->placed << endl;
            cout << "\n\n";
        }
        else{
            cout << "Not Found!!\n\n";
        }
}

void sll::update(){
        Node *p = NULL;
        int no,c;
        bool flag=false;
        p = head;
        cout << "Enter roll no. to be updated: ";
        cin >> no;
        while(p!=NULL){
            if(p->rollno == no){
                flag = true;
                break;
            }
            p = p->next;
        }
        if(flag){
```

```
                cout << "\nThe student with roll no. " << no << " found!\n\nDetails: \n";
                cout << "Roll no: " << p->rollno << endl;
                cout << "Name: " << p->name << endl;
                cout << "PRN: " << p->prn << endl;
                cout << "Placed: " << p->placed << endl;
                cout << "\n\n";
                do{
                        cout << "Enter what you want to update: \n";
                        cout << "1 - Roll no\n2 - Name\n3 - PRN\n4 - Placed\n5 - Cancel\nEnter: ";
                        cin >> c;
                        switch(c){
                                case 1:
                                        cout << "Enter Roll no: ";
                                        cin >> p->rollno;
                                        cout << "Roll no is updated!\n\n";
                                        break;
                                case 2:
                                        cout << "Enter Name: ";
                                        cin >> p->name;
                                        cout << "Name is updated!\n\n";
                                        break;
                                case 3:
                                        cout << "Enter PRN: ";
                                        cin >> p->prn;
                                        cout << "PRN is updated!\n\n";
                                        break;
                                case 4:
                                        cout << "Enter is the student is placed:(0 - false/ 1 - true): ";
                                        cin >> p->placed;
```

```
                                        cout << "Placed is updated!\n\n";
                                        break;
                                case 5:
                                        break;
                                default:
                                        cout << "Enter a valid choice!\n\n";
                        }
                        cout << "Updated data: \n";
                        cout << "Roll no: " << p->rollno << endl;
                        cout << "Name: " << p->name << endl;
                        cout << "PRN: " << p->prn << endl;
                        cout << "Placed: " << p->placed << endl;
                        cout << "\n\n";
                }while(c != 5);
        }
        else{
                cout << "Not Found!!\n\n";
        }
}

void sll::sort(){
        Node *p,*s;
        p = head;
        int roll;
        long long int prn;
        string name;
        float placed;
        while(p!=NULL){
                s = p->next;
                while(s!=NULL){
```

```
                    if (p->rollno > s->rollno){
                        roll = p->rollno;
                        p->rollno = s->rollno;
                        s->rollno = roll;

                        name = p->name;
                        p->name = s->name;
                        s->name = name;

                        prn = p->prn;
                        p->prn = s->prn;
                        s->prn = prn;

                        placed = p->placed;
                        p->placed = s->placed;
                        s->placed = placed;

                        s = s->next;
                    }
                    else{
                        s = s->next;
                    }
                }
                p=p->next;
            }
            cout << "Linked List is sorted as: \n";
            display();
            cout << "\n\n";
}
```

```
void sll::del(){
	int ch;
	Node *p = NULL;
	Node *temp = NULL;
	do{
		cout << "\nEnter: \n";
		cout<<"1 - Delete First one\n2 - Delete Last one\n3 - Delete at specific position\n4 - Delete by specific value\n5 - Cancel\nEnter: ";
		cin >> ch;
		switch(ch){
			case 1:
				p = head;
				head = p->next;
				delete(p);
				cout << "\nFirst Detail is deleted successfully!!\n\n";
				display();
				cout << "\n\n";
				break;
			case 2:
				p = head;
				while(p->next != NULL){
					temp = p;
					p = p->next;
				}
				temp->next = NULL;
				delete(p);
				cout << "\nLast Detail is deleted successfully!!\n\n";
				display();
				cout << "\n\n";
				break;
			case 3:
```

```
                int pos;
                cout  << "\nEnter Position of which the data should be deleted:
";
                cin >> pos;
                p = head;
                for(int i=1; i<pos-1; i++){
                        p = p->next;
                }
                temp = p->next;
                p->next = p->next->next;
                delete(temp);
                cout << "\nData at  position " << pos << " is deleted
successfully!!\n\n";
                display();
                cout << "\n\n";
                break;
        case 4:
                int no,c;
                c = 0;
                bool flag;
                flag = false;
                p = head;
                temp = head;
                cout << "\nEnter Roll no of the student that is to be deleted: ";
                cin >> no;
                while(p != NULL){
                        if(p->rollno==no){
                                flag = true;
                                break;
                        }
                        p = p->next;
```

```
                    c++;
                }
                for(int i=1; i<c; i++){
                    temp = temp->next;
                    if(!flag){
                        cout << "Element not Found!!!\n\n";
                    }else{
                        if(p==head){
                            head = p->next;
                            delete(p);
                        }else if(p->next == NULL){
                            temp = temp->next;
                            delete(p);
                        }else{
                            temp -> next = p->next;
                            delete(p);
                        }
                    }
                    cout << "\nThe Student data with roll no: " << no << "
deleted successfully!!\n\n";
                    display();
                    cout << "\n\n";
                }

                    break;
            case 5:
                    break;
            default:
                    cout << "Enter valid choice!!\n\n";
        }
    }while(ch != 5);
```

```
}

int main(){
	sll l;
	int choice;
	cout << "Welcome to Placement Inforfation System!\n\n";
	cout << "Enter Student Details: \n";
	l.create();
	do{
		cout << "\n\nEnter:\n1 - Display\n2 - Insert\n3 - Reverse\n4 - Search\n5 - Update\n6 - Sort\n7 - Delete\n8 - Exit\n: ";
		cin >> choice;
		switch (choice){
			case 1:
				l.display();
				break;
			case 2:
				l.insert();
				break;
			case 3:
				l.reverse();
				break;
			case 4:
				l.search();
				break;
			case 5:
				l.update();
				break;
			case 6:
				l.sort();
				break;
```

```
                case 7:
                        l.del();
                        break;
                case 8:
                        break;
                default:
                        cout << "Enter valid choice!\n\n";
        }
    }while(choice!=8);

    return 0;
}
```